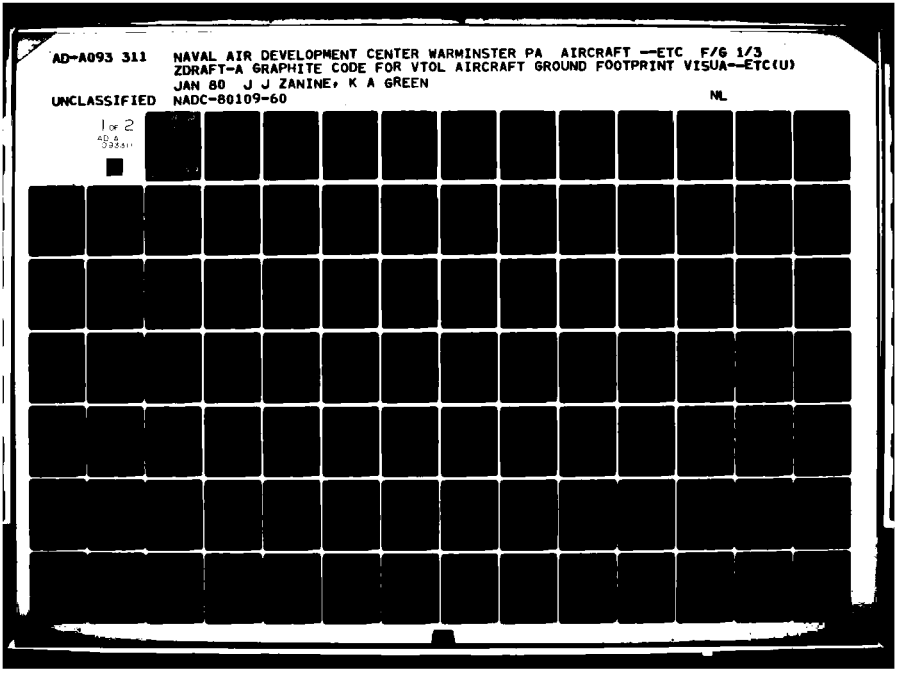
AD-A093 311 NAVAL AIR DEVELOPMENT CENTER WARMINSTER PA AIRCRAFT --ETC F/G 1/3
ZDRAFT-A GRAPHITE CODE FOR VTOL AIRCRAFT GROUND FOOTPRINT VISUA--ETC(U)
JAN 80 J J ZANINE, K A GREEN

UNCLASSIFIED NADC-80109-60 NL

1 OF 2
AD-A
093311



REPORT NO. NADC-80109-60

LEVEL II

12

AD A093311

ZDRAFT - A GRAPHIC CODE FOR VTOL AIRCRAFT GROUND FOOTPRINT VISUALIZATION

J. J. Zanine and K. A. Green
Aircraft and Crew Systems Technology Directorate
NAVAL AIR DEVELOPMENT CENTER
Warminster, Pennsylvania 18974

JANUARY 1980

FINAL REPORT
AIRTASK NO. A3303300/001C/9W058200100
Work Unit No. RB402

Approved for Public Release;
Distribution Unlimited

Prepared for
NAVAL AIR SYSTEMS COMMAND
Department of the Navy
Washington, DC 20361

NOTICES

REPORT NUMBERING SYSTEM - The numbering of technical project reports issued by the Naval Air Development Center is arranged for specific identification purposes. Each number consists of the Center acronym, the calendar year in which the number was assigned, the sequence number of the report within the specific calendar year, and the official 2-digit correspondence code of the Command Office or the Functional Directorate responsible for the report. For example: Report No. NADC-78015-20 indicates the fifteenth Center report for the year 1978, and prepared by the Systems Directorate. The numerical codes are as follows:

CODE	OFFICE OR DIRECTORATE
00	Commander, Naval Air Development Center
01	Technical Director, Naval Air Development Center
02	Comptroller
10	Directorate Command Projects
20	Systems Directorate
30	Sensors & Avionics Technology Directorate
40	Communication & Navigation Technology Directorate
50	Software Computer Directorate
60	Aircraft & Crew Systems Technology Directorate
70	Planning Assessment Resources
80	Engineering Support Group

PRODUCT ENDORSEMENT - The discussion or instructions concerning commercial products herein do not constitute an endorsement by the Government nor do they convey or imply the license or right to use such products.

APPROVED BY: E. J. Sturm DATE: 24 Sept 80

E. J. STURM
CDR USN

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(14) REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: NADC-80109-60

2. GOVT ACCESSION NO.: AD-A093311

3. RECIPIENT'S CATALOG NUMBER:

(6) 4. TITLE (and Subtitle): ZDRAFT-A Graphic Code for VTOL Aircraft Ground Footprint Visualization.

(9) 5. TYPE OF REPORT & PERIOD COVERED: Final Report Oct 78-Jan 80

6. PERFORMING ORG. REPORT NUMBER:

(10) 7. AUTHOR(s): J. J. Zanine and K. A. Green

8. CONTRACT OR GRANT NUMBER(s):

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Aircraft & Crew Systems Technology Directorate
Naval Air Development Center (Code 6052)
Warminster, PA 18974

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS:
AIRTASK A3303300/001C/
910058200100
W.U. RB402

11. CONTROLLING OFFICE NAME AND ADDRESS:
Naval Air Systems Command
Department of the Navy
Washington, DC 20361

(11)

12. REPORT DATE: January 1980

13. NUMBER OF PAGES: 108

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office):

(12) 112

15. SECURITY CLASS. (of this report): Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

(16) W0582

18. SUPPLEMENTARY NOTES

(17) W0582001

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Interactive Computer
Graphics

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The computer program, entitled "ZDRAFT," generates a graphic display of VTOL aircraft and their associated ground flow fields. The actual flow field data is calculated by another computer program. The "ZDRAFT" computer code rapidly assimilates and displays this flow field data. The

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

393522

display consists of pertinent flow field characteristics, such as stagnation lines, upwash flow and ground plane wall jet conditions, superimposed over a scaled aircraft planform. This visual form allows easy assessment of various configurations and operating conditions.

TABLE OF CONTENTS

LIST OF FIGURES

FORWARD

This effort was performed during the period October 1978 to January 1980 and was sponsored by Mr E. Lichtman of the Naval Air Systems Command under Air Task No. A3303300/001C/9W0582001.

SUMMARY

This report documents a computer program, entitled "ZDRAFT", that generates a graphic display of VTOL aircraft and their associated ground flow fields. The actual flow field data is calculated by another computer program. The "ZDRAFT" computer code rapidly assimilates and displays this flow field data. The display consists of pertinent flow field characteristics, such as stagnation lines, upwash flow and ground plane wall jet conditions, superimposed over a scaled aircraft planform. This visual form allows easy assessment of various configurations and operating conditions.

1.0 DESCRIPTION

1.1 Background

The "ZDRAFT" computer graphics program, described in this report, was developed to complement a computer program that estimates the inlet temperature rise of VTOL aircraft operating in ground effect (Reference a). The "ZDRAFT" code was designed to display the ground plane flow fields and the stagnation lines calculated by the other program. This display is most useful in examining parametric variations of aircraft height, nozzle pressure ratios, nozzle vector and splay angles and nozzle aspect ratios. These parametric variations can produce drastic changes in the shape and the location of ground plane and/or undersurface stagnation lines. Hence, a rapid means for visualizing these changes is necessary, to efficiently evaluate conceptual designs. Although, flow field visualization was the primary reason for developing this graphics capability, "ZDRAFT" was constructed for use as a general graphics program. For example, drawings have been developed for aircraft spotting and engine installation studies.

1.2 Program Description and Capabilities

The "ZDRAFT" code is compiled in Fortran IV and uses the Tektronix PLOT10 library. Computer runs are made from a remote interactive graphics terminal on the Naval Air Development Center's (NAVAIRDEVCEN) Central Computer System. This computer system consists of two CDC 6600's and one CDC CYBER 170 Model 175. The "ZDRAFT" code exceeds the memory size limit of the NAVAIRDEVCEN interactive system. Therefore, it is necessary to run the program with segmentation loading. This uses less computer memory by loading only the program segments that are need at a given time.

The specific interactive hardware, necessary for running the code, are:

1. Tektronix 4010-series computer display terminal
2. Tektronix 4953/4954 graphics tablet
3. Tektronix 4631 hard copy unit

The interactive computer display terminal has a keyboard, similar to a conventional typewriter, and a direct storage cathode ray tube (DSCRT) display screen. It is via this keyboard and display screen that the operator communicates with the Central Computer System. The major components of the graphics tablet are a large (1.079M x 0.864M) flat writing surface (tablet) and writing pen or position cursor. The tablet is a magnetically prebiased surface on which paper (film, etc.) may be placed. Under the surface of the tablet is a X-Y grid of magneto-strictive wires, that sense the position of the cursor or writing pen. Hence, a point on the tablet can be converted to a digitial position, and this information transmitted to the display terminal and/or the computer. The hard copy unit allows the operator to make a permanent high contrast copy of the display screen image. The copies are made on special dry silver paper. For specific information concerning this equipment, see References b, c and d.

An interactive computer terminal permits direct interplay between the operator and the computer. The "ZDRAFT" computer program utilizes this interactive capability. The program monitors and supervises the operator by writing directions and questions to the terminal display screen. The operator controls execution and responds to questions with inputs from the cueboard, the terminal keyboard and/or the the graphics tablet. The cueboard is a program generated operation menu which lists the individual program functions (see Figure 1). Selection of a cueboard command causes the specified function to execute. A command may be input to the computer in two ways. First, using the tablet cursor, a selection can be made from a cueboard mounted on the graphics tablet. Second, the cueboard code for the desired command can be typed from the terminal keyboard. The cueboard is the primary instrument for running the "ZDRAFT program. Therefore, by selecting the appropriate cueboard commands and responding to questions and directions, all of the the following proceedures can be duplicated.

Reproduction of a simple line drawing on the terminal display screen is a primary function of the "ZDRAFT" program. After loading the program and answering an initial set of questions, the operator can elect to construct a drawing. In response, the code requests the operator to secure a drawing

to the graphics tablet and to input position, size and scale. This information initializes the graphics tablet and the display screen. Now, the operator can attempt to represent the drawing with a series of line segments. A line segment is a set of points connected by straight lines or with a spline fit curve. For a specific line segment, the operator selects the type of fit and inputs the points from the tablet drawing. These line segments may be displayed as they are completed. For drawings symmetric about the x-axis, the operator can, at his or her option, construct half of the drawing and instruct the code to generate the mirror image. The code simply produces a negative y image of the drawing data and adds to the existing drawing data matrix. This capability is useful with scaled aircraft planforms.

When the drawing is complete, the boundaries of the drawing displayed on the screen can be changed. To accomplish this, the operator redefines the boundries of the graphics tablet drawing. The operator can increase or decrease the area of the drawing shown on the display screen. The scale of the screen drawing will change to utilize the entire display screen. Any of these screen drawings can be saved and retrieved at another time. That is, the drawing data matrix is written to a specified local file, which can be made permanent after the program terminates. To revive a drawing, the data file must be local. At the operator's request, this local file is read into the program.

Additional "ZDRAFT" features include, drawing a border, drawing a x-y axis, calculating and drawing a scale legend, and adding a title. Drawing the border or axis only requires the operator to specify the cueboard command. When selecting the scale legend option, the operator positions or repositions the legend on the drawing. Once the position is fixed, the operator must input the command to draw the legend. The scaling is determined by the computer and will change, if the drawing size is altered. The legend's position, however, relative to the display screen is unaffected by a change in the drawing boundaries. Finally, requesting a title initiates a series ofq set questions which allows the operator to type in lines of a drawing title. The display screen drawing will resize to accommodate the title which is automatically centered over the drawing. This title is written on the screen at the operator's request.

Another basic function of "ZDRAFT" program is the superposition of a VTOL aircraft's ground plane and upwash flow field (i.e. stagnation lines and isocontour lines) over

an appropriate aircraft planform drawing. This stagnation and isocontour information is generated by the hot gas reingestion computer program, "REINGST" (see Reference a) and stored on permanenBefore loading "ZDRAFT" on the computer, the operator makes the specific data file a local file. When running "ZDRAFT" this data file is read by the program at the operator's request. With the planform drawn on the display screen, the operator can select to plot the stagnation lines and/or isocontour lines over the planform.

1.3 Program Structure

The graphics computer program, "ZDRAFT", consists of twenty one subroutines and functions. These subroutines are listed below in alphabetical order.

CAPTION	CPTION
CUEBRD	REVIVE
DRAFT	SAVE
DRAWCUE	SETCUE
DRAWISC	SETSCR
ERASE	SETTD
FIT	STAGLN
INPUT	SYMBOL
ISO	SYMMET
LINES	VIEW
MESSAGE	

In addition, two computer libraries are accessed by "ZDRAFT". These are the Tektronix PLOT10 library and the in-house TEKLIB2 library (see Reference e). A simplified flow chart of the computer code is shown in Figure 3.

1.4 Subroutine Descriptions

CAPTION - handles the input, storage and writing of drawing titles. These titles are input from the terminal keyboard by the operator.

CUEBRD - transforms the operator's cueboard input into calls to the appropriate subroutine.

DRAFT - is the main program in "ZDRAFT". It initializes Tektronix PLOT10 and "ZDRAFT" routines and presets various variables, switches and matrices. Further, the operator is permitted to obtain a copy of cuebcard and/or position the cueboard on the graphics tablet. Operational control of the program is switched to the cueboard.

DRAWCUE - draws a cueboard key on the terminal display screen and generates a hard copy.

DRAWISO - draws the ground plane isocontour lines (e.g., temperature, velocity, etc.) which can be superimposed over an aircraft planform.

ERASE - sets the matrices that contain line segment information to zero. This erases the drawing data from memory, but does not alter the present drawing size, coordinate system or scale.

FIT - controls and orders the spline fit of line segments that describe the aircraft planform.

INPUT - recieves positional inputs from the terminal display screen or the graphics tablet and/or alpha-numeric information from the terminal keyboard. This information is transformed into digital data that is usable by other subroutines.

ISO - reads the isocontour data from a specified file and computes the necessary spline curves.

LINES - controls the input of line segments which describe an aircraft planform. These line segments are a series of points that are connected by a linear or spline curve fit.

MESSAGE - controls all program messages written to the terminal display screen.

OPTION - performs several drawing functions. These include drawing a border, drawing x-y axis and positioning and drawing

a scale legend.

REVIVE - reconstructs a previously generated planform configuration from data on a specified input file.

SAVE - allows the operator to place planform data on a local file. This local file can be made permanent or copied to a magnetic tape for long term storage.

SETCUE - permits the operator to position a copy of the cuebcard key on the graphics tablet.

SETSCR - sets screen limits and scale factors necessary to generate a planform drawing on the terminal display screen.

SETTD - allows the operator to position a drawing on the graphics tablet and describe a specific coordinate system and scale factor.

STAGLN - reads stagnation line data from a specified file and draws the stagnation lines. These stagnation lines can be superimposed over an aircraft planform.

SYMBOL - generates the symbol used in plotting the stagnation line data points.

SYMMET - duplicates existing line segments but with negative y values. These new segments are added to the existing segment producing an image which is symmetric about the x-axis.

VIEW - translates the line segment data into a line drawing on the terminal display screen.

2.0 Conclusions

The graphical code "ZDRAFT" is a useful conceptual design tool for visualizing the primary flow field characteristics of a VTOL aircraft hovering in ground effect. "ZDRAFT" can quickly construct a scaled computer drawing of an aircraft planform from a line drawing. Major flow field characteristics of the VTOL aircraft, such as stagnation lines, upwash flow and ground flow conditions can be rapidly superimposed over this scaled computer planform. Flow field conditions and stagnation line locations for input to the "ZDRAFT" code are generated from experimental data and/or computations from another computer code. "ZDRAFT" has been used extensively to compare experimental data, supplied by industry or taken from technical reports, with computer generated results.

Although this computer code was written specifically for VTOL aircraft ground footprint studies, it was designed to be easily modified for application to other tasks. To date, "ZDRAFT" has been used to scale aircraft planforms for spotting studies and to help analyze aircraft/engine interface problems, at the conceptual design stage.

A01	A02	A03	A04	A05	A06	A07	A08	A09	A10
MESSAGES	SQ/ON,OFF	DRAWCUE	SETCUE	SETDRAWING	SETSCREEN	BORDER	DRAWAXIS	CLEAR	HARDCOPY
B01	B02	B03	B04	B05	B06		B08	B09	
NEWLINE	SPLINE	LINEAR	DRAWALL	DRAW	X-SYM		TITLE	WRITETITLE	
C01	C02	C03	C04	C05	C06	C07	C08	C09	
SAVE	REVIVE	ERASE	STAGTAPE	DRAWSTAG	POSITIONKEY	DRAWKEY	ISOTAPE	DRAWISO	
									D10
									STOP

FIGURE 1 - Tablet Cueboard Key

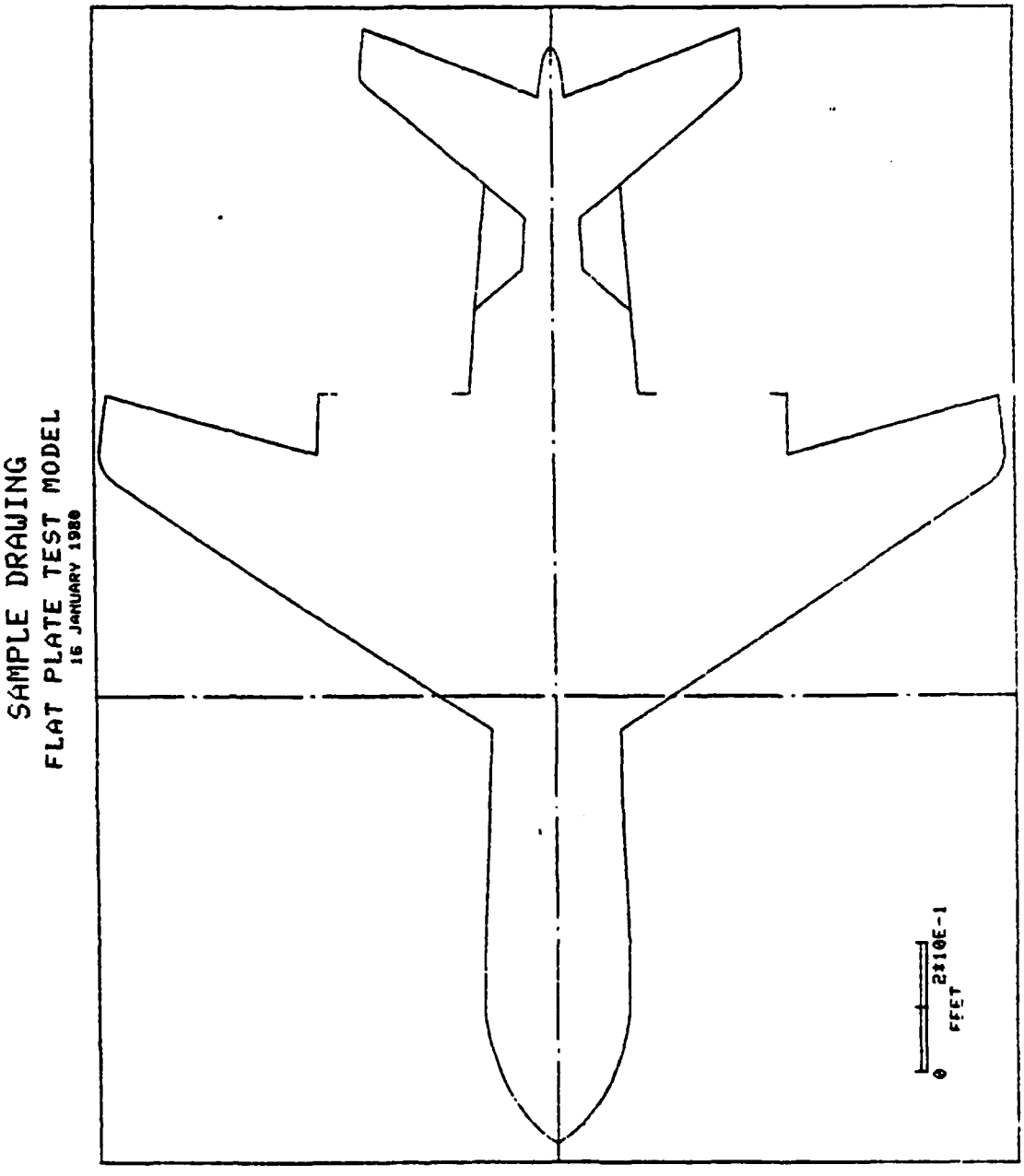


FIGURE 2 - Example of planform drawing.

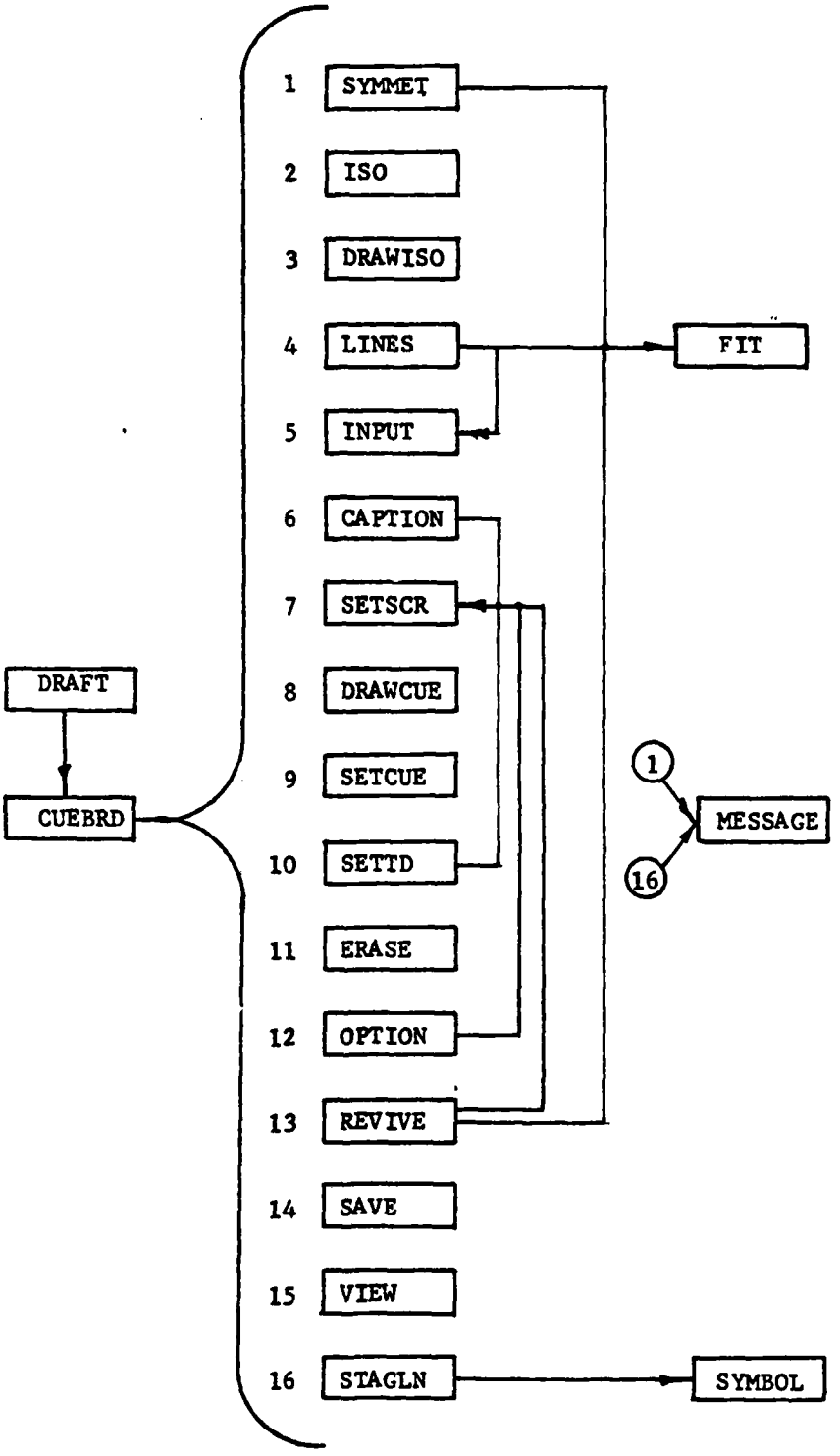


Figure 3 - ZDRAFT Flow Diagram

3.0 References

a) Green, K.A., Zanine, J.J., "Computerized Method for an Estimation of Hot Gas Reingestion for a VTOL Aircraft at the Conceptual Design Stage," NAVAIRDEVCEN Report No. 78256-60.

b) Tektronix 4015 and 4015-1 Computer Display Terminal, Users Instruction Manual, Tektronix, Inc. Beaverton, Oregon.

c) Tektronix 4953/4954, Graphics Tablet, Users Instruction Manual, Tektronix Inc. Beverton, Oregon.

d) Tektronix 4631, Hard Copy unit, Users Instruction Manual, Tektronix Inc. Beverton, Oregon.

e) Caddy, M. J., "TREAD/TLOOK- Multipurpose Computer Routine for Interpolation and Extrapolation of Tabular Data," NAVAIRDEVCEN Report No. 76366-30, 1977.

APPENDIX A - User's Guide

The "ZDRAFT" computer program is not a computational computer code. The purpose of "ZDRAFT" is to facilitate the visualization of ground flow fields generated by VTOL aircraft. The flow field data is produced by another computer program, "REINGST" (Reference a). The output from "REINGST" is maintained on permanent file in a format acceptable to the "ZDRAFT" code. In describing the use and operation of "ZDRAFT", it will be assumed that the stagnation line and isocontour data stored on permanent files, will be retrieved to local files.

"ZDRAFT" is run from a interactive graphic computer terminal. The interactive capability allows the operator direct control over program execution. Further, this capability allows the computer code to monitor operator inputs and to veto improper requests. In "ZDRAFT, the operator controls the program through an operation cueboard and a question and answer format. The cuebcard is a checkerboard drawing with key words and three digit alpha-numeric codes identifying the individual blocks (see Figure 1). This drawing is taped to the graphics tablet, and its position is input to the program with the tablet cursor. When control is switched to the cueboard, the operator can select a specific block with the tablet cursor. Alternately, control can be passed to the terminal keybcard and the corresponding alpha-numeric code can be entered via the keyboard. This will cause the computer to execute the specified action. Examples of these actions are clearing the display screen, drawing an aircraft planform, requesting the input of data points from the tablet, terminating the program, reading and plotting stagnation line data, generating an informative message on the display screen, etc. An explanation of each cueboard command is given in section (A.1).

(A.1) Explanation of Cueboard Commands

The following commands are available to the operator via the tablet cueboard or using the terminal keyboard codes shown in parenthesis:

MESSAGES (A01) - acts as an on/off switch for questions/instructions that normally would appear on the terminal display screen. Error messages, however, are not suppressed and will appear if something is done incorrectly by the operator.

SC/CN,OFF (A02) - switches control from the graphics tablet cueboard to the terminal keyboard and display screen, or vice versa.

DRAWCUE (A03) - clears the display screen and constructs a cueboard.

SETCUE (A04) - permits the operator to position or reposition the cueboard on the graphics tablet.

SETDRAWING (A05) - allows the operator to secure a drawing to the graphics tablet and to specify size, scale and axis system.

SETSCREEN (A06) - sets screen limits and scale factors necessary to generate a drawing on the terminal display screen.

BORDER (A07) - draws a border around the display screen drawing.

DRAWAXIS (A08) - draws the x-y axis on the display screen drawing. The axes are drawn with a dash-dot line.

CLEAR (A09) - clears the display screen. No information is cleared from memory.

HARDCOPY (A10) - produces a hard copy of current screen information.

NEWLINE (B01) - sets the program to create a line segment

unrelated to previous segments. Next, the operator must select the LINEAR or the SPLINE cue.

SPLINE (BC2) - permits a series of points from the drawing to be input and fitted with a spline curve. (NOTE: Values of X must be increasing.) Any other cueboard command ends the line segment.

LINEAR (BC3) - permits a series of points from the drawing to be input and fitted with straight lines. Any other cueboard command ends the line segment.

DRAWALL (BC4) - draws all existing line segments on the display screen.

DRAW (BC5) - draws line segments on display screen starting from the end of last line segment drawn.

X-SYM (BC6) - Causes the computer to symmetrically reproduce the drawing about the X-axis.

TITLE (BC8) - permits the operator to input, from the terminal keyboard, drawing titles. The title is input one line at a time, and the character size is specified for each line. There are four character sizes defined by the integers from 1 to 4. The largest size is 1 and the smallest is 4.

WRITETITLE (BC9) - causes the drawing title to be written on the display screen. The drawing will be resized and repositioned to accommodate the title. This is not automatic, however, and the drawing should be reconstructed after the title is written.

SAVE (CC1) - writes the existing planform data to a local file. The operator must input the tape number which corresponds to the local file. Tape numbers from 5 to 15 are valid.

REVIVE (CC2) - reads in a previously saved planform from a local file. The operator must input the tape number which corresponds to the local file. Tape numbers from 5 to 15 are valid.

ERASE (CC3) - sets all line segment data to zero. This erases the line segment data from memory, but does not alter present drawing size, coordinate system, scale, or actual drawing already existing on the screen.

STAGTAPE (C04) - allows the operator to specify the tape number which corresponds to a local file containing stagnation line data. Tape numbers from 5 to 15 are valid.

DRAWSTAG (C05) - reads the stagnation line tape previously specified and plots the data to the same scale as the planform drawing.

POSITIONKEY (C06) - allows operator to position the scale bar legend at a convenient location on the drawing.

DRAWKEY (C07) - determines the proper size for the scale legend and draws it on the screen at the specified location.

ISOTAPE (C08) - allows isocontour data to be read from a local file and performs the computations necessary to spline fit this data. The operator must input the tape number which corresponds to the local file. Tape numbers from 5 to 15 are valid.

DRAWISO (C09) - draws the previously compiled isocontour lines on the screen drawing.

STOP (D10) - terminates the program.

(A.2) Drawing Planform Configuration

The "ZDRAFT" computer code has the capability of computerizing a simple line drawing. That is, the code allows the user to transform a line drawing into a set of variables and a matrix of points. Having transformed the drawing into numerical data, the code interprets the information and produces a line drawing on the terminal display screen. Further, the code can alter the size and scale, draw a border and a x-y axis, generate a scale legend, add titles, superimpose other drawings, etc. These features are used in the construction of an aircraft planform.

Assume the initial steps in running "ZDRAFT" are complete. The process of computerizing a scaled aircraft planform is initiated by selecting the cueboard command, SETDRAWING. The initial steps are predetermined and instructions are output to the display screen. First, the operator is directed to secure the drawing on the graphics tablet and using the tablet cursor, specify the lower left and upper right corners of the drawing. Next, the operator is requested to specify two points on the drawing and to input their x-y values. Finally, the operator must designate the units of linear dimensions. With this information the code sets scaling factors and initializes the display screen. At this point, control is returned to the operator and construction of planform drawing can start.

The aircraft planform is described by a series of line segments. These line segments are defined by a set of points which are connected by either straight lines or a spline fit curve. To begin construction of the planform drawing the cueboard command NEWLINE is selected. This primes the code for the input of in a new line segment. The operator is then requested to select the LINEAR or the SPLINE command. This determines the fit for the following set of drawing points. Using the tablet cursor, the operator selects points from the planform drawing. Selection of any cueboard block terminates this set of points. The LINEAR or SPLINE command selected after a set of points, will initiate another set of points. The first point of this second set is identical to the last point of the preceding set. To input a line segment unconnected to preceding segments, NEWLINE must be the first

command. When complete the drawing data can be written to a local file by specifying SAVE. This file can be made permanent after termination of the program.

The following information details the exact input and response to generate a computer drawing with a bar scale legend, title information, a border, and an axis system. Examples of the drawing partially complete are also given in Figures A1 through A5. Note, the symbol, "cr",indicates that a carriage return must be input from the terminal keyboard.

type.... CALL,PABS9. "cr"

(screen clears)

prompt.. do you wish a copy
of the graphics tablet cueboard?
type y(yes) or n(no), and "cr".

type.... Y "cr"

prompt.. is hard copier sufficiently warm?
type y(yes) or n(no), and "cr".

type.... Y "cr"

(cueboard is drawn on the terminal display screen
and a hardcopy is produced)

prompt.. do you wish to position
the graphics tablet cueboard?
type y(yes) or n(no), and "cr".

type.... Y "cr"

prompt.. input lower left and lower right
corners of cueboard?

(secure copy cueboard on the graphics tablet
and input points with tablet cursor)

(cueboard is now operative)

cuebrd.. SETDRAWING

(use the tablet cursor to make selection from cueboard)

prompt.. place drawing on the graphics tablet.

(secure drawing on graphics tablet)

prompt.. input lower left and upper right
corners of drawing.

(input points with cursor)

prompt.. specify point on drawing
and type in (x,y) coordinates.

(input point with cursor)

type.... 0.0 , 0.0 "cr"

prompt.. specify another point
and type in (x,y) coordinates.

(input point with cursor)

type.... 1.2125 , 0.0 "cr"

prompt.. type unit of dimensions (20 characters or less)

type.... FEET "cr"

(construction of aircraft planform can now begin)

cuebrd.. NEWLINE

prompt.. new line started select linear or spline.

cuebrd.. SPLINE

(use tablet cursor to select points from drawing on tablet,
points will be connected with a spline fit,
points must have increasing values of x
with a minimum of three points)

(for the present example, start at the aircraft nose,
input points)

cuebrd.. SPLINE

(starts a new spline curve that is connected to the previous curve, input points)

cuebrd.. LINEAR

(select points to be connected with straight lines, lines connect with preceding curve, input points)

cuebrd.. SPLINE

(starts new spline fit that is connected to preceding curve, input points)

cuebrd.. LINEAR

(starts another series of straight lines, input points)

cuebrd.. DRAW

(draws all curves input, since last draw command)

cuebrd.. HARDCOPY

(see Figure A1)

cuebrd.. NEWLINE

prompt.. new line started select linear or spline.

cuebrd.. LINEAR

(starts a series of straight lines not connected to the preceding curve, input points)

cuebrd.. NEWLINE

prompt.. new line started select linear or spline.

cuebrd.. LINEAR

(input points)

cuebrd.. SPLINE

(input points)

cuebrd.. DRAW

cuebrd.. HARDCOPY

(see Figure A2)

cuebrd.. X-SYM

(this generates a negative y image of the existing
drawing data, producing an image symmetric
about the x-axis)

cuebrd.. DRAW

cuebrd.. HARDCOPY

(see Figure A3)

cuebrd.. SAVE

type.... 5 "cr"

(places drawing data on tape5, this local file
can be made permanent)

cuebrd.. POSITIONKEY

prompt.. set scale position.

(using the tablet cursor, select a position on the drawing
for the scale legend)

cuebrd.. DRAWKEY

(the scale legend is drawn on the display screen)

cuebrd.. HARDCOPY

(see Figure A4)

cuebrd.. CLEAR

(screen clears)

cuebrd.. TITLE

```
prompt..    input title lines.  character is preset to 3.

 (four character sizes are available, 1 through 4,
  larger numbers represent smaller character sizes)

prompt..    input character size.
            eof default is previous character size.

type....  1 "cr"

prompt..    type title line

type....  SAMPLE DRAWING "cr"

prompt..    sample drawing

prompt..    is this line correct?
            type y(yes) or n(no), and "cr".

type....  Y "cr"

prompt..    input character size.
            eof default is previous character size.

type....  2 "cr"

prompt..    type title line

type....  FLAT PLATE TEST MODEL "cr"

prompt..    flat plate test model

prompt..    is this line correct?
            type y(yes) or n(no), and "cr".

type....  Y "cr"

prompt..    input character size.
            eof default is previous character size.

type....  4 "cr"

prompt..    type title line

type....  16 JANUARY 1980 "cr"

prompt..    16 january 1980
```

prompt.. is this line correct?
type y(yes) or n(no), and "cr".

type.... Y "cr"

prompt.. input character size.
eof default is previous character size.

type.... "cr"

prompt.. type title line

type.... "cr"

(a return of carriage when a title line is requested,
ends title line inputs)

(screen clears and title is written)

cuebrd.. HARDCOPY

(see Figure A5)

cuebrd.. BORDER

(border is drawn)

cuebrd.. DRAWAXIS

(x-y axis is drawn)

cuebrd.. DRAWKEY

(scale legend is drawn)

cuebrd.. DRAWALL

(planform is drawn)

cuebrd.. HARDCOPY

(see Figure A6)

(A.3) Superimposing Stagnation and Isocontour Lines

This example demonstrates the superposition of stagnation lines over a specific aircraft planform (see Appendix B.1). The stagnation line data was previously generated by "REINGST" (Reference a) and stored on permanent file or magnetic tape. The previously developed aircraft planform was also stored on permanent file or tape. To be read by the "ZDRAFT" program, these data files must be placed on local files. Appropriate local file names are TAPE5 throught TAPE15. Assume the aircraft planform data is on TAPE5, the ground plane stagnation line data is on TAPE6 and the undersurface stagnation line data is on TAPE7. After repeating the initial inputs, the following commands will generate a drawing of the planform and the stagnation lines.

cuebrd.. STAGTAPE

prompt.. input tape number that contains stagnation line data.

type.... 6 "CR"

cuebrd.. CLEAR

(screen clears)

cuebrd.. BORDER

(border is drawn)

cuebrd.. DRAWALL

(aircraft planform is drawn)

cuebrd.. DRAWAXIS

(x-y axis is drawn)

cuebrd.. DRAWSTAG

(stagnation lines are plotted)

cuebrd.. HARDCOPY

(see Figure A7)

cuebrd.. STOP

prompt.. stop requested from cueboard.

(execution is terminated)

Figure A7 illustrates the stagnation line data generated by the computer program, "REINGST" (Reference a). "REINGST" produces a stagnation line for each sequential pair of nozzles. For a simple two nozzle case, stagnation lines for nozzle pairs 1-2 and 2-1 are generated. These stagnation lines would be identical. In the complex six nozzle system, stagnation lines are calculated for nozzle pairs 1-2, 2-3, 3-4, 4-5, 5-6 and 6-1. Because "REINGST" does not compensate for the interference between stagnation lines, portions of the stagnation lines are imaginary. These imaginary sections can be determined by plotting the stagnation lines. The section of a stagnation line plotted before the intersection with another line is imaginary. Complex nozzle configurations may have a series of stagnation line intersections. Determining the governing intersection may be difficult.

In the present illustration the three stagnation lines intersect at a single point on the x-axis (see Figure A7). Hence, locating the non-existent portion of the stagnation lines is relatively simple. The stagnation line of nozzle pair 1-2 plots from top to bottom; therefore, the section above the x-axis is imaginary. Similarly, the section of the 3-1 stagnation line below the x-axis must be eliminated. For the nozzle pair 2-3, the segement to the left of the intersection should not exist. Figures A8 and A10 show a portion of the uncorrected data file for stagnation lines 1-2 and 2-3, respectively. Using the text editor, the imaginary sections can be deleted from the data files. The revised data files are shown in Figures A9 and A11.

This problem of imaginary stagnation line segments, also, occurs with the undersurface stagnation lines. Determination of the imaginary portions of these lines may prove more difficult than the ground plane lines. Therefore, the undersurface line data should be revised after the ground plane data. The ground plane revisions can then be used as a

guide in eliminating the non-existent undersurface sections.

The corrected stagnation line data is on local files TAPE6 and TAPE7. The isocontour data is taken from permanent file and placed on local file, TAPE10. Now, a final drawing of the aircraft planform with superposed stagnation lines and isocontour lines can be constructed. Note, the cueboard commands are now input from the terminal keyboard.

type.... CALL,PABS9. "cr"

(screen clears)

prompt.. do you wish a copy of the cueboard?
type y(yes) or n(no), and "cr".

type.... Y "cr"

prompt.. do you wish to position the cueboard?
type y(yes) or n(no), and "cr".

type.... N "cr"

prompt.. input cue codes from terminal keyboard.

(cue codes are shown on the cueboard)

type.... C2 "cr"

(cue - revive)

prompt.. input tape number that contains drawing data.

type.... 5 "cr"

type.... B8 "cr"

(cue - title)

prompt.. input title lines. character is preset to 3.

(four character sizes are available, 1 through 4,
larger numbers represent smaller character sizes)

prompt.. input character size.

```
                eof default is previous character size.

type....  1 "cr"

prompt..    type title line.

type....  SAMPLE DRAWING "cr"

prompt..    sample drawing
            is this line correct?
            type y(yes) or n(no), and "cr".

type....  Y "cr"

prompt..    input character size.
            eof default is previous character size.

type....  2 "cr"

prompt..    type title line.

type....  FLAT PLATE TEST MODEL "cr"

prompt..    flat plate test model

prompt..    is this line correct?
            type y(yes) or n(no), and "cr".

type....  Y "cr"

prompt..    input character size.
            eof default is previous character size.

type....  4 "cr"

prompt..    type title line.

type....  16 JANUARY 1980 "cr"

prompt..    16 january 1980

prompt..    is this line correct?
            type y(yes) or n(no), and "cr".

type....  Y "cr"

prompt..    input character size.
            eof default is previous character size.
```

```
type....  "cr"

prompt..    type title line.

type....  "cr"

 (a return of carriage when a title line is requested,
  ends title line inputs)

 (screen clear and title is written))


type....  A7 "cr"

 (cue - border,
  border is drawn)

type....  A8 "cr"

 (cue - drawaxis,
  x-y axis is drawn)

type....  B4 "cr"

 (cue - drawall,
  aircraft planform is drawn)

type....  A1 "cr"

 (cue - messages,
  screen messages will not be printed,
   except error messages)

type....  C4 "cr"

 (cue - stagtape,
  stagnation line data tape number is requested)

type....  6 "cr"

type....  C5 "cr"

 (cue - drawstag,
  stagnation lines are printed)

type....  C4 "cr"

 (cue - stagtape,
```

```
  stagnation line data tape number is requested)

type....  7 "cr"

type....  C5 "cr"

 (cue - drawstag,
  stagnation lines are plotted)

type....  C6 "cr"

 (cue - positionkey,
  screen cursors appear,
  use screen cursors to postiton scale legend)

type....  P

 (do not return carriage)

type....  C7 "cr"

 (cue - drawkey,
  scale legend is drawn)

type....  A10 "cr"

 (cue - hardcopy,
  see Figure A12)

type....  A9 "cr"

 (cue - clear,
  screen clears)

type....  A1 "cr"

 (cue - messages,
  screen messages are turned on)

type....  C8 "cr"

 (cue - isotape)

prompt..    input tape number containing iso data.

type....  10 "cr"

prompt..    this is a test isocontour line tape
```

read this data?
type y(yes) or n(no), and "cr".

type.... Y "cr"

type.... C4 "cr"

(cue - stagtape)

prompt.. input tape number that contains stagnation
line data.

type.... 6 "cr"

type.... A9 "cr"

(cue - clear,
screen clears)

type.... A8 "cr""

(cue - drawaxis,
x-y axis is drawn)

type.... A7 "CR"

(cue - border,
border is drawn)

type.... B9 "cr"

(cue - writetitle,
title is written)

type.... B4 "cr"

(cue - drawall,
aircraft planform is drawn)

type.... A1 "cr"

(cue - messages,
screen messages will not be printed,
except error messages)

type.... C5 "cr"

(cue - drawstag,

```
  stagnation lines are plotted)

type....  C9 "cr"

 (cue - drawiso,
  isocontour lines are drawn)

type....  C6 "cr"

 (cue - positionkey,
  screen cursors appear,
  use screen cursors to postiton scale legend)

type....  P

 (do not return carriage)

type....  C7 "cr"

 (cue - drawkey,
  scale legend is drawn)

type....  A10 "cr"

 (cue - hardcopy,
  see Figure A13)

type....  D10 "cr"

 (cue - stop)

prompt..    stop requested from cueboard

 (execution is terminated)
```

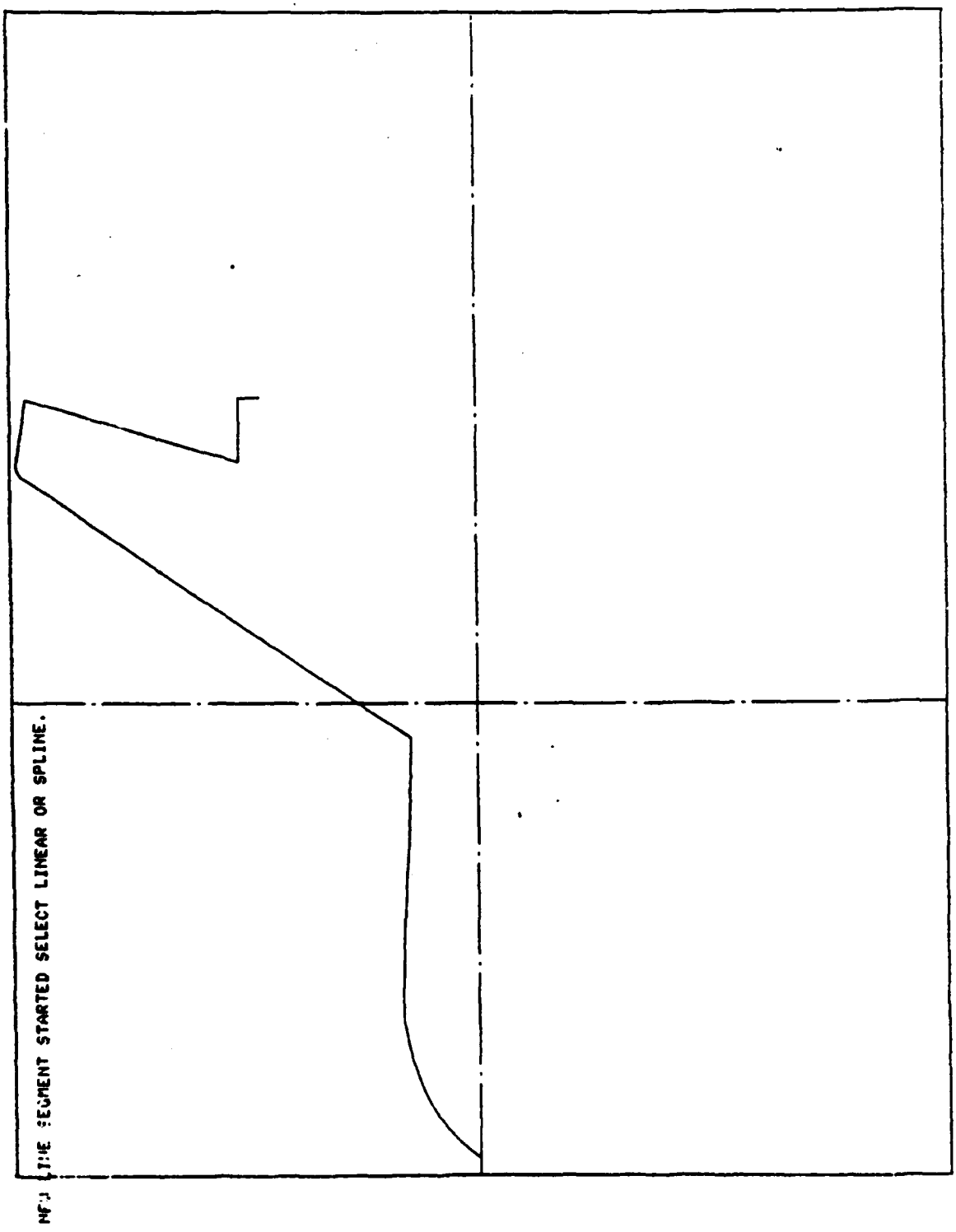


FIGURE A1 - Partial Drawing

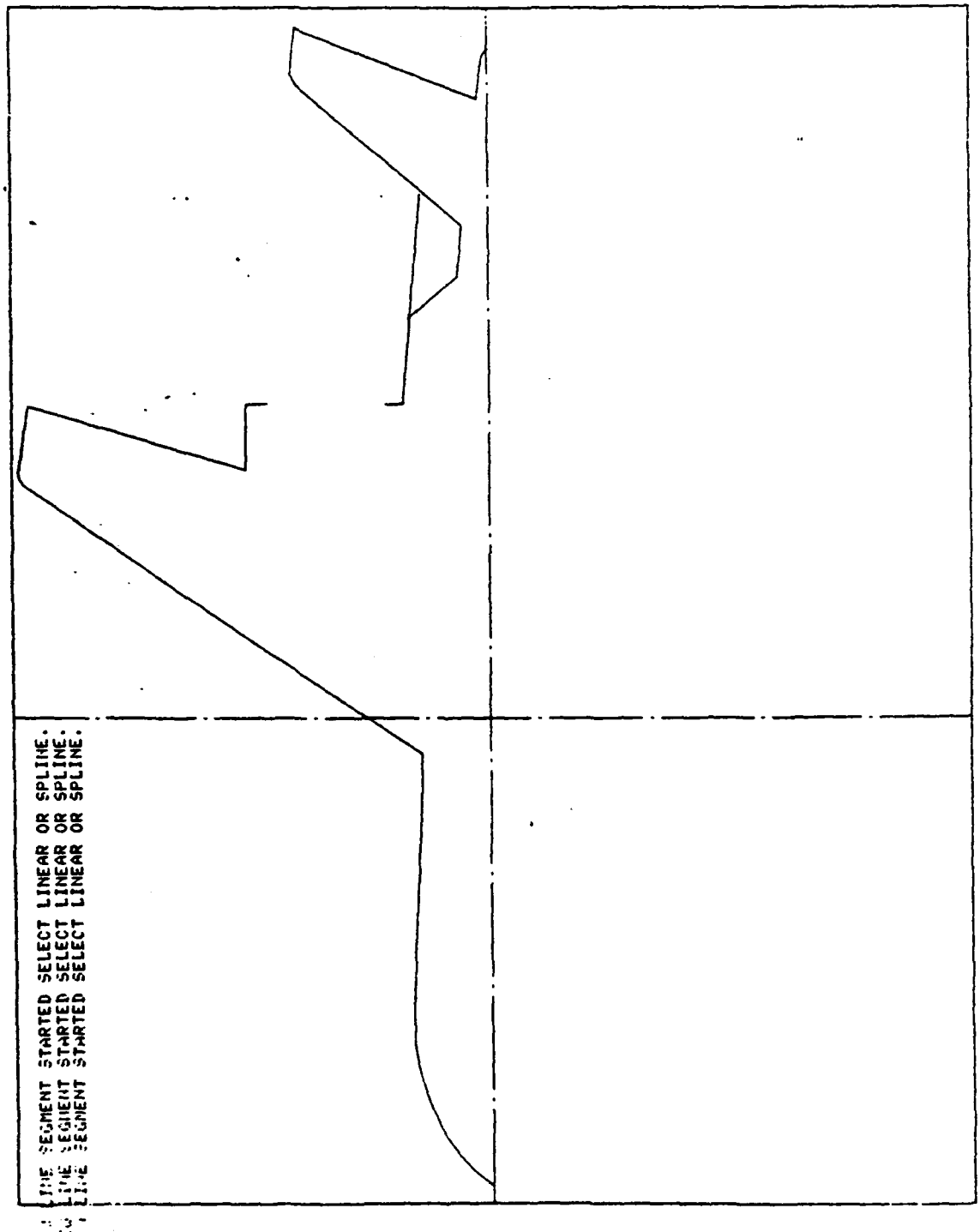


FIGURE A2 - Upper half of planform completed.

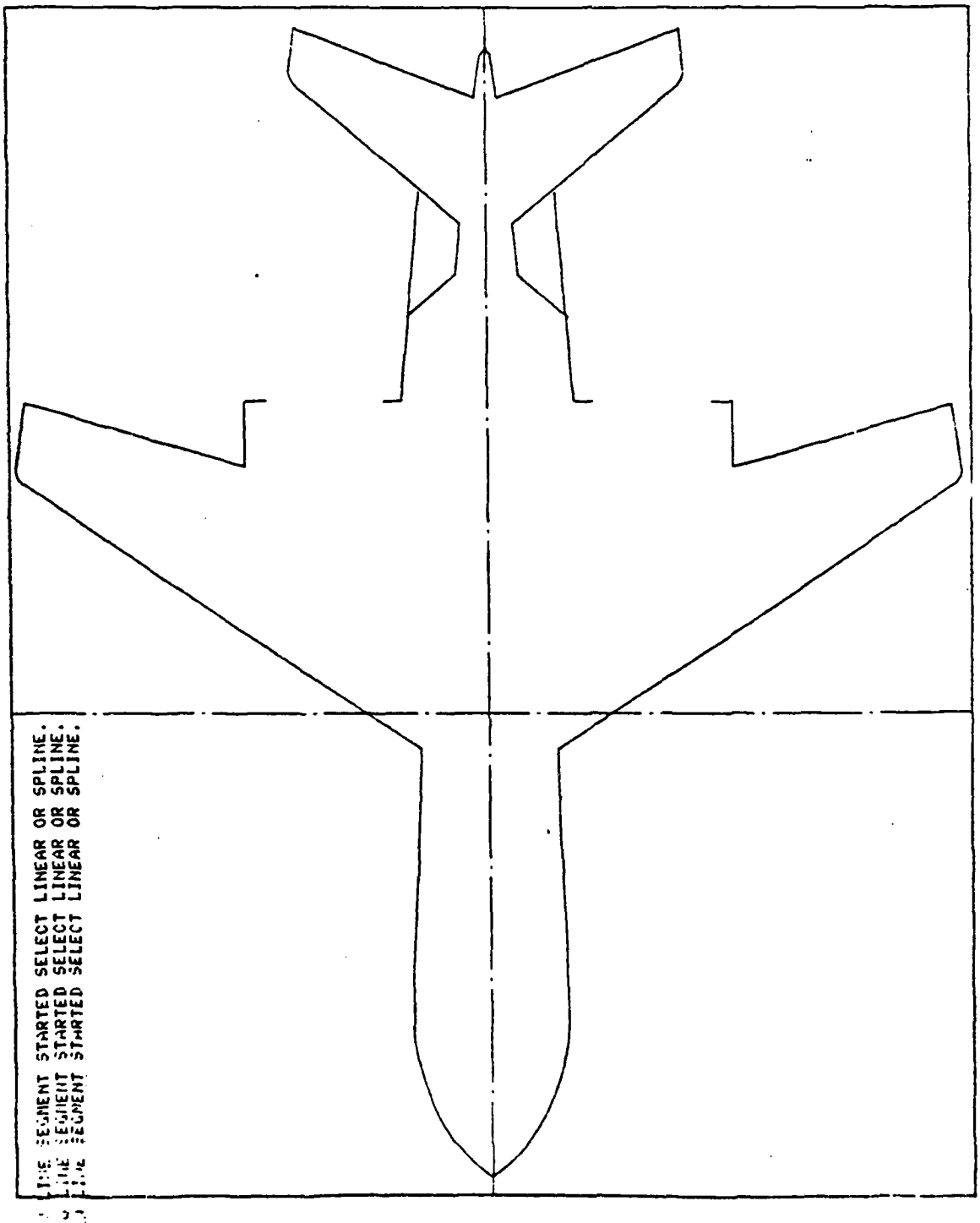


FIGURE A 3 - Symmetric planform automatically produced about x-axis by subroutine "SYMMET".

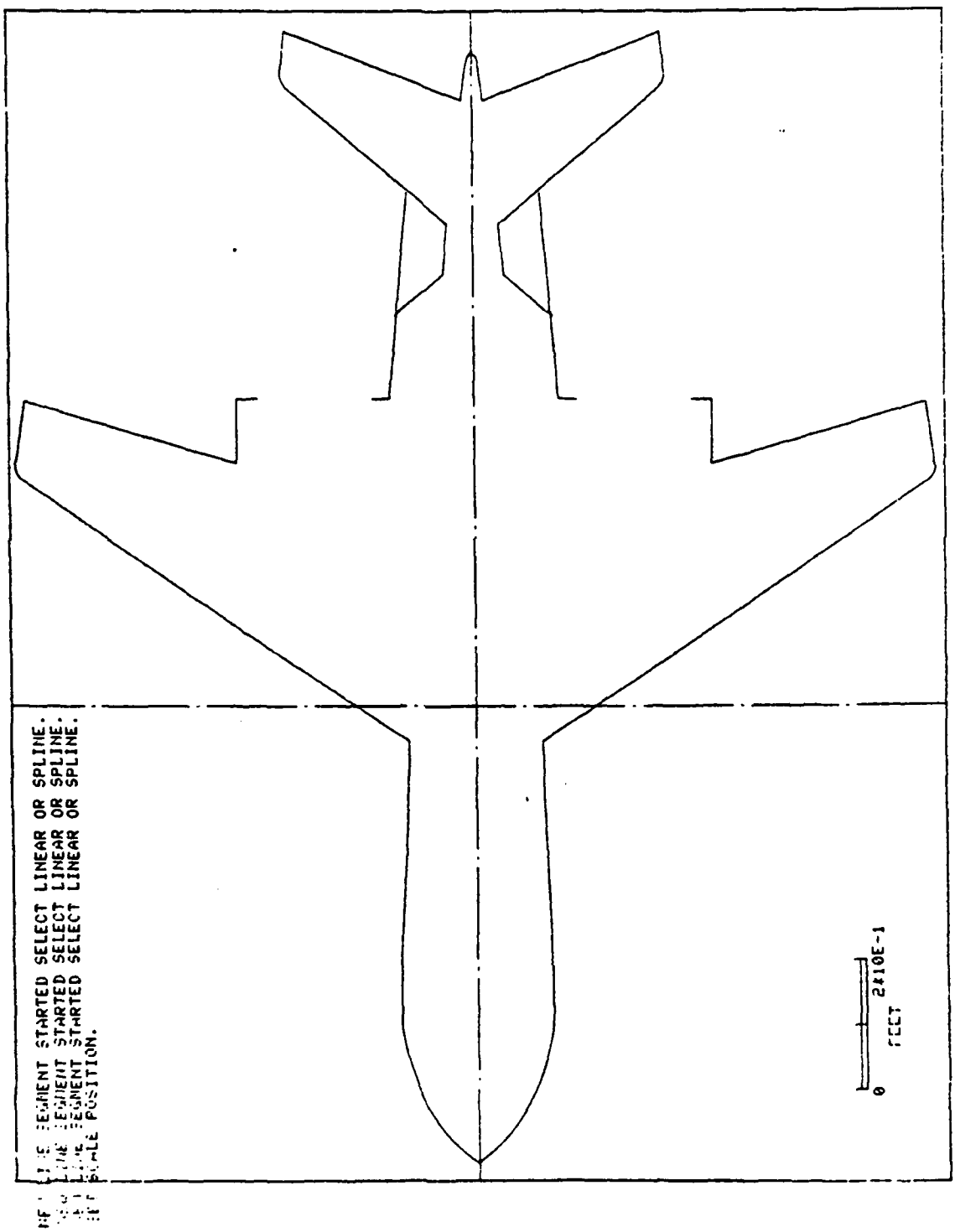


FIGURE A4 - Scale positioned and drawn.

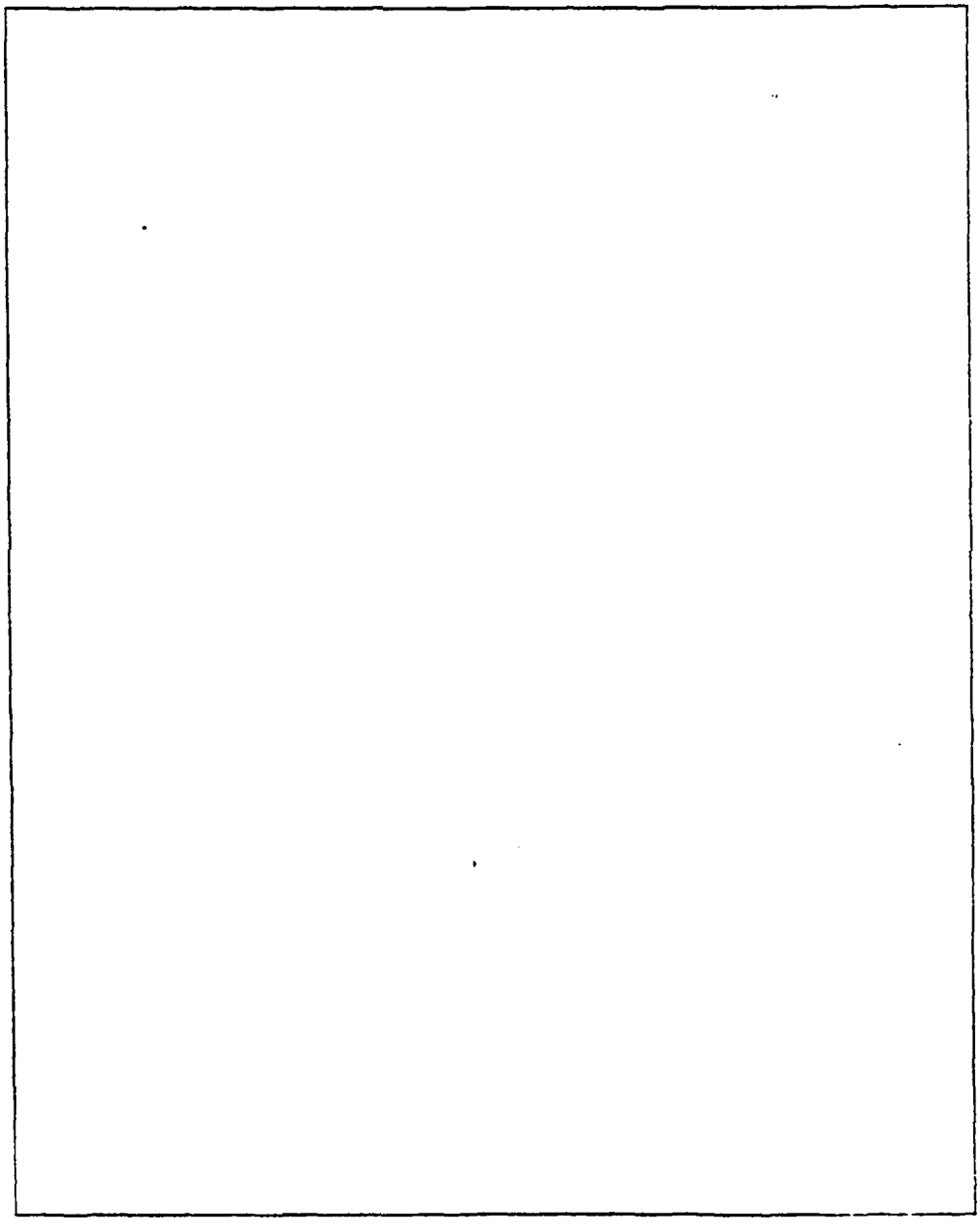


FIGURE A5 - Title information developed and drawn.

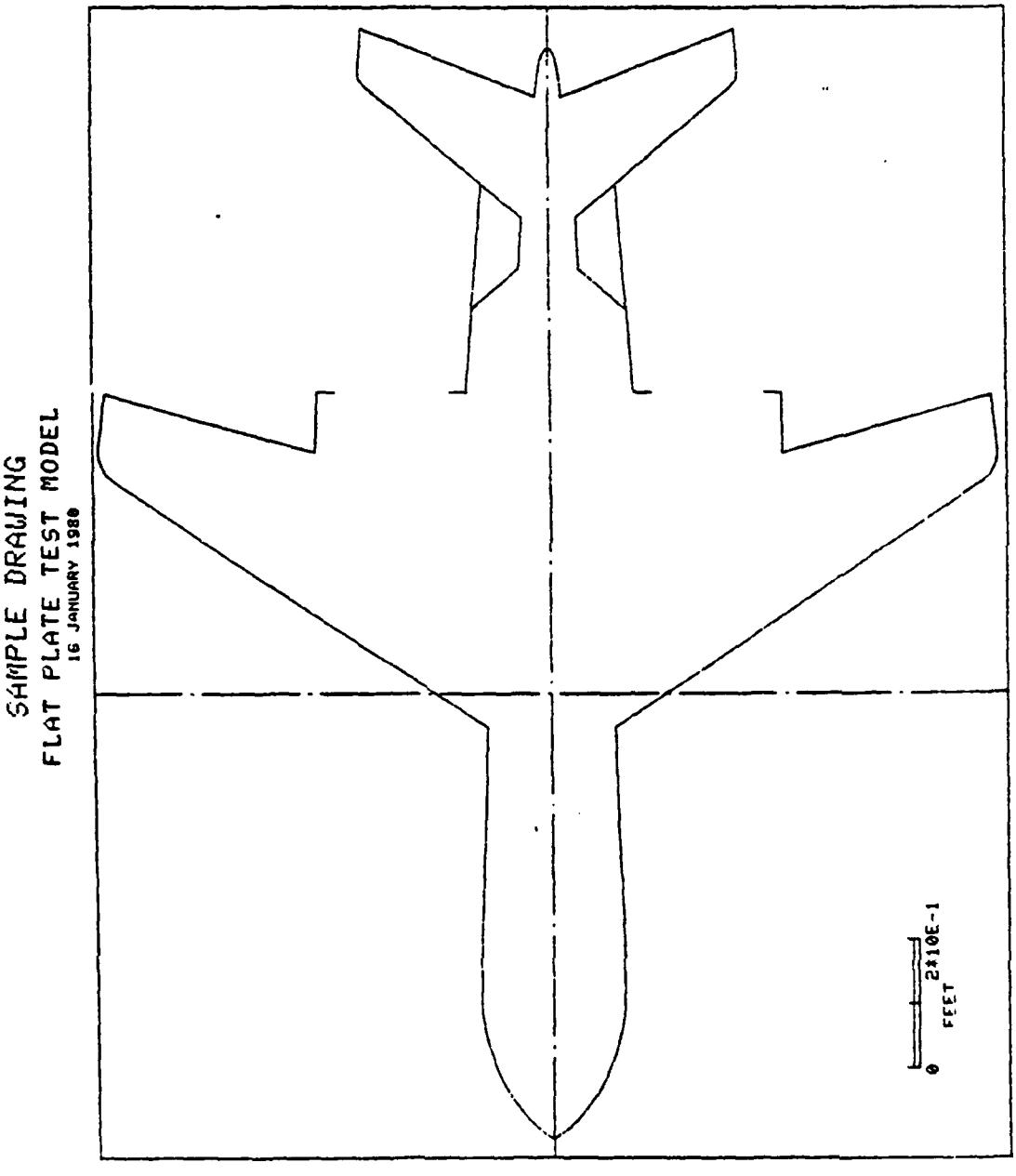


FIGURE A6 - Planform drawing completed.

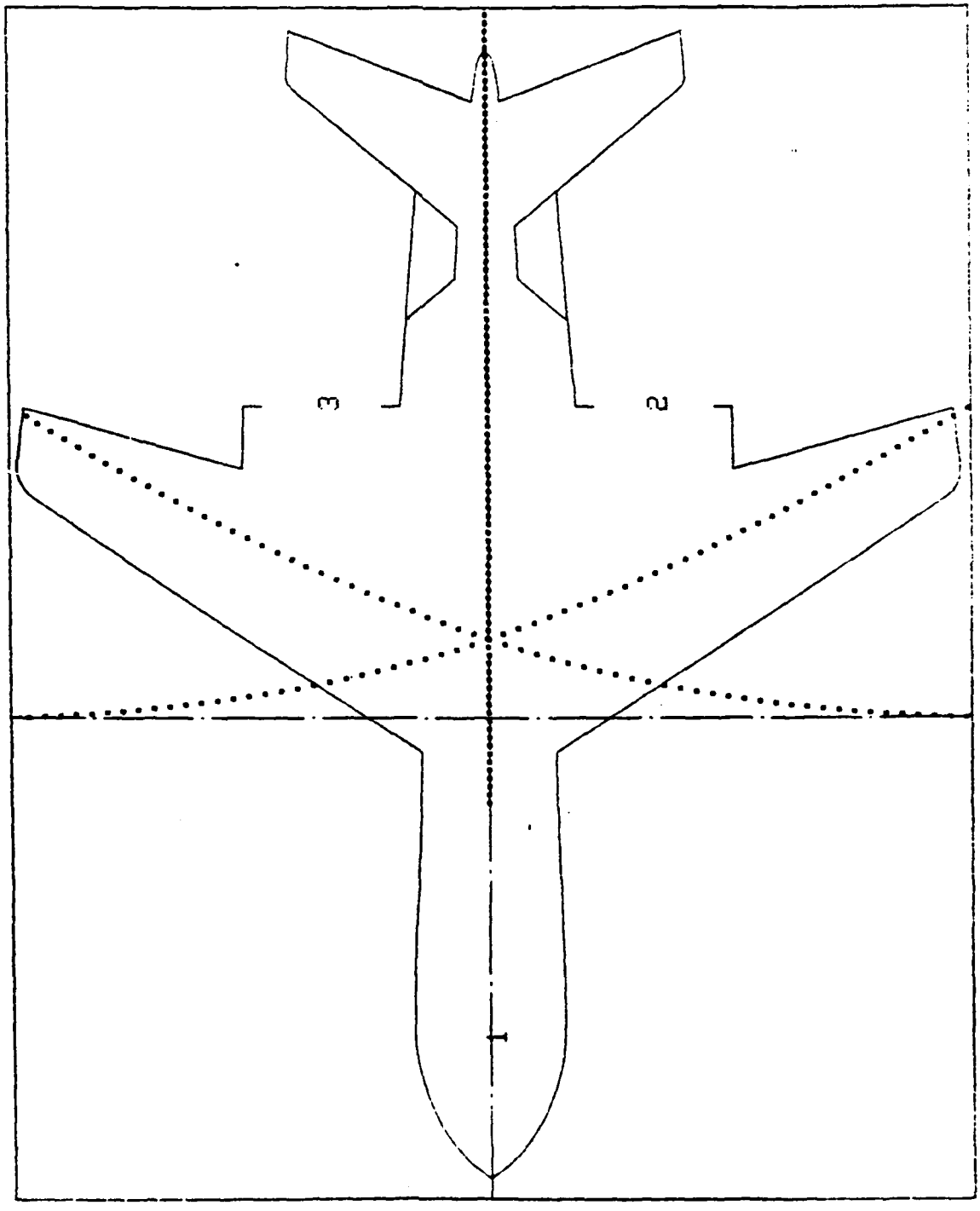


FIGURE A7 - Planform drawing showing impingement points and all ground plane stagnation line data.

DATA POINT	X VALUE	Y VALUE	
-1	.597700	0.000000	Impingement points
-2	-.597700	-.309170	
-3	-.597700	.309170	
10	-.738070	1.184937	
20	-.722552	1.158253	Locate on this line.
30	-.707071	1.131547	
40	-.691628	1.104820	
50	-.676224	1.078069	
60	-.660862	1.051295	
70	-.645544	1.024496	
80	-.630271	.997670	
90	-.615046	.970818	
100	-.599872	.943937	
110	-.584750	.917026	
120	-.569685	.890084	
130	-.554678	.863109	
140	-.539733	.836100	
150	-.524853	.809055	
160	-.510041	.781972	
170	-.495302	.754850	
180	-.480640	.727686	
190	-.466058	.700479	
200	-.451561	.673226	
210	-.437155	.645926	
220	-.422844	.618575	
230	-.408635	.591172	
240	-.394532	.563713	
250	-.380543	.536197	
260	-.366675	.508619	Drop all of these points.
270	-.352933	.480978	
280	-.339327	.453270	
290	-.325865	.425492	
300	-.312555	.397641	
310	-.299406	.369713	
320	-.286428	.341706	
330	-.273632	.313614	
340	-.261029	.285436	
350	-.248629	.257168	
360	-.236445	.228806	
370	-.224489	.200347	
380	-.212773	.171789	
390	-.201311	.143128	
400	-.190115	.114361	
410	-.179199	.085488	
420	-.168575	.056505	
430	-.158256	.027413	
440	-.148254	-.001790	
450	-.138581	-.031103	
460	-.129246	-.060526	
470	-.120258	-.090057	
480	-.111626	-.119693	
490	-.103354	-.149433	
500	-.095448	-.179271	
510	-.087896	-.209201	
520	-.080712	-.239222	
530	-.073893	-.269327	
540	-.067436	-.299512	
550	-.061335	-.329772	
560	-.055583	-.360099	
570	-.050173	-.390490	
580	-.045092	-.420937	
590	-.040333	-.451436	
600	-.035881	-.481981	
610	-.031726	-.512569	
620	-.027855	-.543193	
630	-.024256	-.573851	
640	-.020915	-.604538	
650	-.017820	-.635251	
660	-.014960	-.665986	

FIGURE A8 - Uncorrected ground plane stagnation line data.
Data for nozzles 1 and 2.

DATA POINT	X VALUE	Y VALUE	
-1	.597700	0.000000	} IMPINGEMENT POINTS
-2	-.597700	-.309170	
-3	-.597700	.309170	
440	-.148254	-.001790	
450	-.138581	-.031103	
460	-.129246	-.060526	
470	-.120258	-.090057	
480	-.111626	-.119693	
490	-.103354	-.149433	
500	-.095448	-.179271	
510	-.087896	-.209201	
520	-.080712	-.239222	
530	-.073893	-.269327	
540	-.067436	-.299512	
550	-.061335	-.329772	
560	-.055583	-.360099	
570	-.050173	-.390490	
580	-.045092	-.420937	
590	-.040333	-.451436	
600	-.035881	-.481981	
610	-.031726	-.512569	
620	-.027855	-.543193	
630	-.024256	-.573851	
640	-.020915	-.604538	
650	-.017820	-.635251	
660	-.014960	-.665986	
670	-.012321	-.696741	
680	-.009893	-.727514	
690	-.007664	-.758302	

FIGURE A9 - Corrected ground plane stagnation line data.
Data for nozzles 1 and 2.

DATA POINT	X VALUE	Y VALUE	
10	.159767	-.000000	Drop all of these points.
20	.144308	-.000000	
30	.128550	-.000000	
40	.113391	-.000000	
50	.097933	-.000000	
60	.082474	-.000000	
70	.067016	-.000000	
80	.051557	-.000000	
90	.036099	-.000000	
100	.020640	-.000000	
110	.005182	-.000000	
120	-.010277	-.000000	
130	-.025735	-.000000	
140	-.041194	-.000000	
150	-.056652	-.000000	
160	-.072111	-.000000	
170	-.087569	-.000000	
180	-.103028	-.000000	
190	-.118486	-.000000	
200	-.133945	-.000000	
210	-.149403	-.000000	
220	-.164862	-.000000	
230	-.180320	-.000000	
240	-.195779	-.000000	
250	-.211237	-.000000	
260	-.226696	-.000000	
270	-.242154	-.000000	
280	-.257613	-.000000	
290	-.273071	-.000000	
300	-.288530	-.000000	
310	-.303988	-.000000	
320	-.319447	-.000000	
330	-.334905	-.000000	
340	-.350364	-.000000	
350	-.365822	-.000000	
360	-.381281	-.000000	
370	-.396739	-.000000	
380	-.412198	-.000000	
390	-.427656	-.000000	
400	-.443115	-.000000	
410	-.458573	-.000000	
420	-.474032	-.000000	
430	-.489490	-.000000	
440	-.504949	-.000000	
450	-.520407	-.000000	
460	-.535866	-.000000	
470	-.551324	-.000000	
480	-.566783	-.000000	
490	-.582241	-.000000	
500	-.597700	-.000000	
510	-.613158	-.000000	
520	-.628617	-.000000	
530	-.644075	-.000000	
540	-.659534	-.000000	
550	-.674992	-.000000	
560	-.690451	-.000000	
570	-.705909	-.000000	
580	-.721368	-.000000	
590	-.736826	-.000000	
600	-.752285	-.000000	
610	-.767743	-.000000	
620	-.783202	-.000000	
630	-.798660	-.000000	
640	-.814119	-.000000	
650	-.829577	-.000000	
660	-.845036	-.000000	
670	-.860494	-.000000	
680	-.875953	-.000000	
690	-.891411	-.000000	
700	-.906870	-.000000	
710	-.922328	-.000000	
720	-.937787	-.000000	
730	-.953245	-.000000	
740	-.968704	-.000000	
750	-.984162	-.000000	

FIGURE A10 - Uncorrected ground plane stagnation line data.
Data for nozzles 2 and 3.

DATA POINT	X VALUE+	Y VALUE
210	-.149403	-.000000
220	-.164862	-.000000
230	-.180320	-.000000
240	-.195779	-.000000
250	-.211237	-.000000
260	-.226696	-.000000
270	-.242154	-.000000
280	-.257613	-.000000
290	-.273071	-.000000
300	-.288530	-.000000
310	-.303988	-.000000
320	-.319447	-.000000
330	-.334905	-.000000
340	-.350364	-.000000
350	-.365822	-.000000
360	-.381281	-.000000
370	-.396739	-

FIGURE A11 - Corrected ground plane stagnation line data
Data for nozzles 2 and 3.

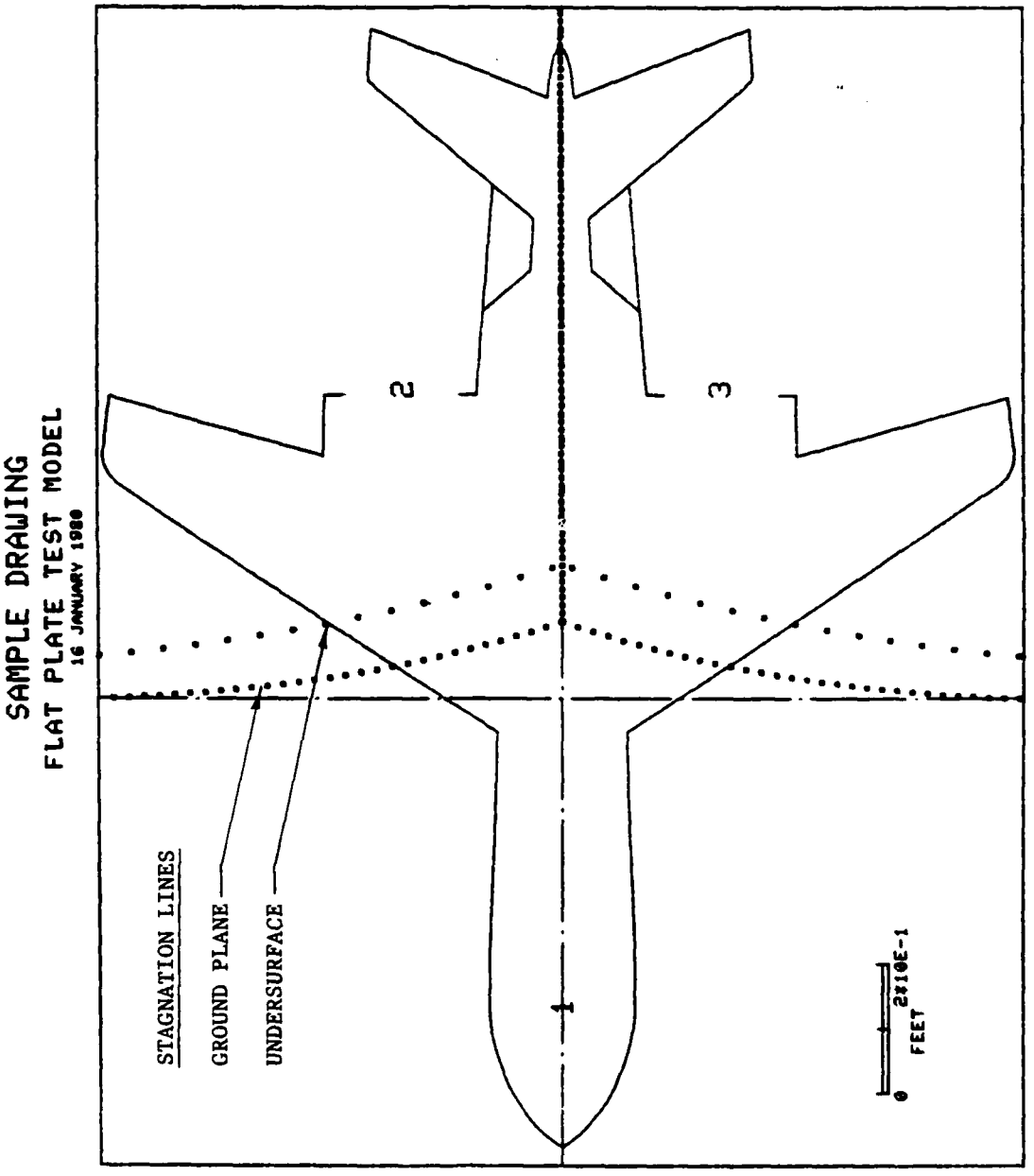


FIGURE A12 - Completed drawing showing ground plane and undersurface stagnation lines.

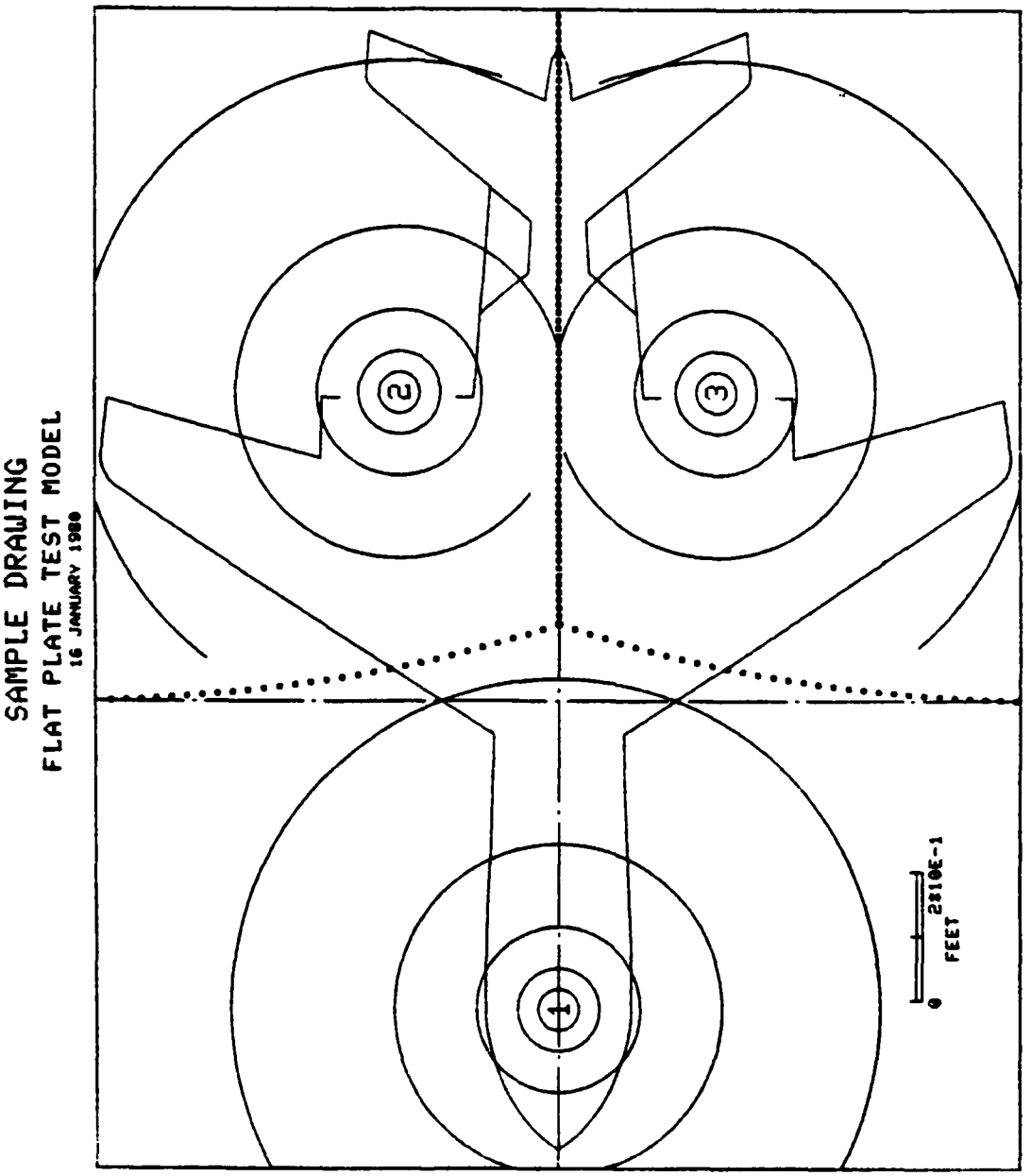


FIGURE A13 - Completed drawing showing ground plane stagnation lines and isocontour lines.

APPENDIX B - "ZDRAFT" COMPUTER PROGRAM

(B.1) Listing of Procedure Files

Various procedure files necessary to compile, segment, load and run the "ZDRAFT" code are listed below. Permanent file, ZDRAFT9, contains the fortran program "ZDRAFT". The following procedure file, PZD9, packs and compiles the fortran code. The fortran compiled version is saved on file, ZD9REL4, and the fortran listing is written to file, OUT.

```
PZD9
GET,ZDRAFT9.
PACK,ZDRAFT9.
FTN4(I=ZDRAFT9,R=3,L=OUT,B=ZD9REL4)
SAVE,ZD9REL4.
```

Next, the proceedure file, PESC9, segments the code and saves the absolute version on ZD9ABS.

```
PSEC9
GET,SEGDIR9,ZD9REL4,XTAPE.
GET,SEGREF,PL1CLIB/UN=SYSTEM.
GET,TEKLIB2/UN=VT1781.
COPYEF,XTAPE,TAPE1.
COPYEF,XTAPE,TAPE2.
REWIND,TAPE1,TAPE2.
MAP,PART.
SEGREF,ZD9REL4,SEGOUT.
SEGLOAD(I=SEGDIR9)
LIBLOAD(PL1CLIB,CHECK)
LIBLOAD(PL1CLIB,LABEL)
LDSET(LIB=PL1CLIB/TEKLIB2/PL1CLIB/TEKLIB2,MAP=/MAPP)
LOAD,ZD9REL4.
NOGO,ZD9ABS.
```

The following is the segmentation load directions.
SEGDIR9.

```
BR1       TREE      DRAWISO
DRAWISO   INCLUDE   ISO,SPLNC1,UPDATE
DRAWISO   GLOBAL    BLKISO-SAVE
BR2       TREE      ERASE
ERASE     INCLUDE   FIT,LINES,REVIVE,SETTD,SPLNC1,SYMMET,UPDATE,VIEW
```

```
BR3       TREE      CAPTION
CAPTION   INCLUDE   OPTION,STAGLN,SYMBOL
CAPTION   GLOBAL    BLKCAPT-SAVE
BR4       TREE      DRAWCUE
DRAWCUE   INCLUDE   SETCUE
TRUNK     TREE      ZDRAFT-(BR1,BR2,BR3,BR4)
ZDRAFT    INCLUDE   CUEBRD,MESSAGE,INPUT
          GLOBAL    BLK,BLKJ,BLKPTS,BLKSCAL,BLKSD,BLKTD,BLKCUE,TKTRNX
          END
```

Finally, the proceedure file to run the segmented absolute version of the "ZDRAFT" computer code.

```
PABS9
GET,ZD9ABS,XTAPE.
COPYBF,XTAPE,TAPE1.
COPYBF,XTAPE,TAPE2.
REWIND,TAPE1,TAPE2.
MAP,PART.
ZD9ABS.
```

(E.2) Listing of the Input File

XTAPE is a multi-file permanent file that contains information input to the "ZDRAFT" code. The first file contains the messages that the code outputs to the terminal display screen. The second contains information pertaining to the structure of the cueboard.

```
TAPE1
 1200 IBAUD, TRANSMISSION RATE - CHARACTERS PER SECOND
 4096 MAXSR, NUMBER OF ADDRESSABLE POINTS
    3 ITERM, TERMINAL IDENTIFICATION NUMBER
    3 ISIZE, CHARACTER SIZE (4014/15 ONLY)
    0 LOCDS, LOCAL DISPLAY OF TABLET POINT ON TERMINAL (0 - NO)(1 - YES)
    0 IPEN,  TABLET MODE (0 - PEN MODE) (1 - PRESENCE MODE)
01 PQ DO YOU WISH A COPY OF THE CUEBOARD ? TYPE Y(YES) OR N(NO), AND "CR".
02 PQ IS HARDCOPIER ON AND SUFFICIENTLY WARM ? TYPE Y(YES) OR N(NO), AND "CR".
03 PQ TURN ON HARDCOPIER. WHEN WARM TYPE Y AND "CR".
04 PQ IS COPY SATISFACTORY ? TYPE Y(YES) OR N(NO), AND "CR".
05 P
06 P  POINTER IMPROPERLY POSITIONED, TRY AGAIN.
07 P  INPUT LOWER LEFT AND LOWER RIGHT CORNERS OF CUEBOARD.
08    INPUT CUE CODES FROM TERMINAL KEYBOARD.
09 PS CUEBOARD DATA NOT FOUND ON TAPE2.
10 PQ DO YOU WISH TO POSITION THE CUEBOARD? TYPE Y(YES) OR N(NO), AND "CR".
11    INPUT TAPE NUMBER ON WHICH DRAWING DATA IS TO BE SAVED.
12 PS TABLET CUEBOARD DATA NOT FOUND ON TAPE3.
13 PS NUMBER OF TABLET CUEBOARD ROWS EXCEEDS MATRIX DIMENSION.
14 PS NUMBER OF TABLET CUEBOARD COLUMNS EXCEEDS MATRIX DIMENSION.
15 PS NUMBER OF TABLET CUEBOARD A/N CHARACTERS EXCEEDS MATRIX DIMENSION.
16 PS CAPTION TO LONG FOR TABLET CUE BLOCK.
17 P  BLANK CUE REQUESTED. TRY AGAIN.
18    PLACE DRAWING ON THE GRAPHICS TABLET.
19    INPUT LOWER LEFT AND UPPER RIGHT CORNERS OF THE DRAWING.
20    SPECIFY POINT ON DRAWING AND TYPE IN (X,Y) COORDINATES.
21    SPECIFY ANOTHER POINT AND TYPE IN (X,Y) COORDINATES.
22 PS DISTANCE BETWEEN SELECTED POINTS CANNOT BE ZERO.
23 P  DRAWING DIMENSIONS HAVE NOT BEEN SPECIFIED.
24 P  INSUFFICIENT NUMBER OF POINTS FOR SPLINE."
25 PQ DO YOU WISH TO ABORT THIS LINE SEGMENT ? TYPE Y(YES) OR N(NO), AND "CR".
26 P  CURRENT LINE SEGMENT ABORTED.
27 P  INPUT ADDTIONAL POINTS FOR SPLINE CURVE.
28    NEW LINE SEGMENT STARTED SELECT LINEAR OR SPLINE.
```

```
30 P  MAXIMUM NUMBER OF POINTS HAS BEEN EXCEEDED.
31 P  MAXIMUM NUMBER OF SPLINES HAS BEEN EXCEEDED.
32 P  TWO CONSECUTIVE NCODES EQUAL 0.
33 P  NCODE IS NOT -5 , -1 , 0 , 1 , 5 .
34 P  NCODE SHOULD BE EQUAL TO -5 .
35 P  VALUES OF X MUST BE INCREASING FOR SPLINE FIT.
36  Q ERASE DRAWING ? TYPE Y(YES) OR N(NO), AND "CR".
37 P  THE CUEBCARD HAS NOT BEEN POSITIONED.
38    INPUT TAPE NUMBER THAT CONTAINS DRAWING DATA.
39 P  EOF ENCOUNTERED READING DRAWING DATA TAPE.
40    INPUT TAPE NUMBER THAT CONTAINS STAGNATION LINE DATA.
41 P  TAPE NUMBER CONTAINING STAGNATION DATA HAS NOT BEEN SPECIFIED.
42 P  SCREEN SCALE IS LESS THAN 1.0*E-10.
43 P  SCREEN SCALE IS GREATER THAN 1.0*E10.
45    SET SCALE KEY POSITION.
46    TYPE UNIT OF DIMENSIONS (20 CHARACTERS OR LESS).
47 P  INPUT CHARACTER SIZE. EOF DEFAULT IS PREVIOUS CHARACTER SIZE.
48 P  TYPE TITLE LINE.
49 PQ IS THIS LINE CORRECT ? TYPE Y(YES) OR N(NO), AND "CR".
50 P  CHANGE CHARACTER SIZE IF DESIRED.
51 P  RETYPE LAST LINE.
52 P  INPUT TITLE LINES. CHARACTER IS PRESET TO 3.
53    INPUT TAPE NUMBER CONTAINING ISO DATA.
54 PQ READ THIS DATA ? TYPE Y(YES) OR N(NO), AND "CR".
55 P  EOF ENCOUNTERED READING ISO DATA TAPE.
56 P  NUMBER OF VARIABLES EXCEEDS MAXIMUM LIMIT OF 20.
57 P  NUMBER OF ANGLES EXCEEDS MAXIMUM LIMIT OF 20.
58 P  NUMBER OF ISO CONTOUR LINES EXCEEDS MAXIMUM LIMIT OF 20.
59 P  IMPROPER CUE CODE. TRY AGAIN.

-EOF-

TAPE2
   04 NUMBER CUEBCARD ROWS
   10 NUMBER CUEBCARD COLUMNS
   20 NUMBER OF CHARACTERS PER CAPTION
A01  MESSAGES              CALL MESFLIP
A02  SC/CN,OFF             CALL CUEFLIP
A03  DRAWCUE               CALL DRAWCUE
A04  SETCUE                CALL SETCUE
A05  SETDRAWING            CALL SETTD
A06  SETSCREEN             CALL SETSCR
A07  BORDER                CALL BORDER
A08  DRAWAXIS              CALL AXIS
A09  CLEAR                 CALL MESSAGE(0,NDUM)
A10  HARDCOPY              CALL HDCOPY
B01  NEWLINE               CALL NEWLINE
B02  SPLINE                CALL SPLINE(NTHBLK),RETURNS(60)
```

```
B02  LINEAR          CALL LINEAR(NTHBLK),RETURNS(60)
B04  DRAWALL         CALL VIEWALL
B05  DRAW            CALL VIEW
B06  X-SYM           CALL SYMMET
B08  TITLE           CALL CAPTION
B09  WRITETITLE      CALL WTITLE
C01  SAVE            CALL SAVE
C02  REVIVE          CALL REVIVE
C03  ERASE           CALL ERASED
C04  STAGTAPE        CALL STAGLN
C05  DRAWSTAG        CALL DRWSTAG
C06  POSITIONKEY     CALL SETKEY
C07  DRAWKEY         CALL KEY
C08  ISOTAPE         CALL ISO
C09  DRAWISO         CALL DRAWISO
D10  STOP            STOP"REQUESTED FROM CUEBCARD"

-EOF-
```

(E.3) Listing of the "ZDRAFT" Computer Code

```
CZDRAFT
      PROGRAM ZDRAFT(INPUT=101,OUTPUT,TAPE3=INPUT,TAPE4=OUTPUT,
     *               TAPE1=101,TAPE2=101,TAPE5=101,TAPE6=101,
     *               TAPE7=101,TAPE8=101,TAPE9=101,TAPE10=101,
     *               TAPE11=101,TAPE12=101,TAPE13=101,TAPE14=101,
     *               TAPE15=101)
      COMMON
     */BLK/
     *  IHORZ(4),
     *  IVERT(4),
     *  IYSET,
     *  MAXCOL,
     *  MAXIX,
     *  MAXIY,
     *  MAXROW,
     *  MAXSR,
     *  NF
      COMMON
     */BLKCUE/
     *  IHEIGHT,
     *  ILENGTH,
     *  ISIDE,
     *  IUQDELT,
     *  IXORIG,
     *  IUQ1,
     *  IVQDELT,
     *  IYORIG,
     *  IVQ1,
     *  LEGEND(15,15,20),
     *  NCHARQ,
     *  PHIQ,
     *  SIDEQ
      COMMON
     */BLKJ/
     *  JCUE,
     *  JDRAW,
     *  JMESS,
     *  JSCR,
     *  JSTAG,
     *  JTITLE
      COMMON
     */BLKPTS/
     *  ARRAY(500,2),
     *  ASPL(100,20),
     *  NCODE(500),
     *  NPTS,
     *  NSPLS(500),
     *  NSWITCH,
```

```
*  NTHSPL,
*  NVPTS,
*  UNIT
 COMMON
*/BLKSCAL/
*  LENSCAL,
*  NEXP,
*  SCALAST,
*  SCALES,
*  SCALET,
*  SCALSIZ
 COMMON
*/BLKSD/
*  IXRANGE,
*  IXSTART,
*  IX1,
*  IYRANGE,
*  IYSTART,
*  IY1,
*  NUNIT(20),
*  PHID,
*  RATIOD,
*  XRANGE,
*  XSTART,
*  XD,
*  YRANGE,
*  YSTART,
*  YD
 COMMON
*/BLKTD/
*  IUDRW1,
*  IUDRW2,
*  IUD,
*  IVDRW1,
*  IVDRW2,
*  IVD
 DATA
*  (ARRAY=3000*0.0),
*  (IUDRW1=6*0),
*  (IUQ1=0),
*  (IVQ1=0),
*  (IX1=-1),
*  (IY1=-1),
*  (JCUE=-1),
*  (JDRAW=0),
*  (JMESS=1),
*  (JSCR=0),
*  (JSTAG=0),
```

```
     *  (LEGEND=4500*55B),
     *  (NCODE=1004*0),
     *  (SCALAST=0.0)
C    THIS ROUTINE INITIALIZES VARIOUS PLOT10 AND ZDRAFT ROUTINES
C
C    VARIABLES READ FROM TAPE1
C    IBAUD .... TRANSMISSION RATE (BAUD),CHARACTERS PER SECOND
C    MAXSR .... NUMBER OF ADDRESSABLE POINTS ON THE TEKTRONIX TERMINAL
C    ITERM .... TERMINAL IDENTIFICATION NUMBER
C                1 - 4006,4010,4012/13
C                2 - 4014/15
C                3 - 4014/15 ENHANCED GRAPHICS MODULE
C    ISIZE .... CHARACTER SIZE, 4014/15 ONLY
C                    CHARS/LINE  NUM LINES
C                1 -     74          35
C                2 -     81          38
C                3 -    121          58
C                4 -    133          64
C    LOCOS .... SWITCH FOR LOCAL DISPLAY OF TABLET POINT ON TERMINAL
C                0 - NO
C                1 - YES
C    IPEN ..... TABLET MODE
C                0 - PEN MODE
C                1 - PRESENCE MODE
C
      REWIND 1
      READ(1,1000)
      READ(1,1000) IBAUD,MAXSR,ITERM,ISIZE,LOCOS,IPEN
      IF(EOF(1).NE.0) STOP"INITIALIZING VARIABLES FOR PLOT10 NOT FOUND O
     *N TAPE1."
C
C    INITT (PLOT10 ROUTINE) INITIALIZES TERMINAL
      IBAUD=IBAUD/10
      CALL INITT(IBAUD)
C
C    TERM (PLOT10 ROUTINE) SPECIFIES TERMINAL TYPE
      CALL TERM(ITERM,MAXSR)
      IF(ITERM.EQ.1) GO TO 200
C
C    THE FOLLOWING STORES ALPHA/NUMERIC CHARACTER DIMENSIONS
      DO 100 ISIZE=1,4
      CALL CHRSIZ(ISIZE)
      CALL GSIZE(IH,IV)
      IHORZ(ISIZE)=IH
      IVERT(ISIZE)=IV
  100 CONTINUE
C
C    TABINT (PLOT10 ROUTINE) INITIALIZES GRAPHICS TABLET
```

```
      ICOORD=MAXSR/4000
      CALL TABINT(ICOORD,LOCOS,IPEN)
C
C   INITIALIZES MESSAGE ROUTINE
  200 CALL MESSAGE(MAXSR,NDUM)
C
C   THE FOLLOWING READS CUE BLOCK SPECIFICATIONS FROM TAPE2
C   AND INITIALIZES CUE BLOCK VARIABLES.
      REWIND 2
      READ(2,2000)
C
C   MESSAGE - "SCREEN CUE BOARD DATA NOT FOUND ON TAPE2"
      IF(EOF(2).NE.0) CALL MESSAGE(9,NDUM)
      READ(2,2100) MAXROW,MAXCOL,NCHARQ
      IF(EOF(2).NE.0) CALL MESSAGE(9,NDUM)
C
C   MESSAGE - "NUMBER OF TABLET CUE BOARD ROWS EXCEEDS MATRIX DIMENSION"
      IF(MAXROW.GT.15) CALL MESSAGE(13,NDUM)
C
C   MESSAGE - "NUMBER OF TABLET CUE BOARD COLUMNS EXCEEDS MATRIX DIMENSION"
      IF(MAXCOL.GT.15) CALL MESSAGE(14,NDUM)
C
C   MESSAGE - "NUMBER OF SPECIFIED CUE BOARD A/N CHARACTERS EXCEEDS MATRIX DIMEN
      IF(NCHARQ.GT.20) CALL MESSAGE(15,NDUM)
      NF=MAXSR/1024
      MAXIX=MAXSR - 1
      MAXIY=NF*780
  360 MAXBLK=MAXROW*MAXCOL
      DO 400 NTHBLK=1,MAXBLK
      READ(2,2200) NROW,NCOL,(LEGEND(NROW,NCOL,N),N=1,NCHARQ)
      IF(EOF(2).NE.0) GO TO 420
  400 CONTINUE
  420 IXORIG=NF*510
      IYORIG=NF*390
      ISIDE=NF*(1000/MAXCOL)
      IDUMMY=NF*(760/MAXROW)
      IF(ISIDE.GT.IDUMMY) ISIDE=IDUMMY
      ILENGTH=MAXCOL*ISIDE
      IHEIGHT=MAXROW*ISIDE
C
C   MESSAGE - "DO YOU WISH A COPY OF THE GRAPHICS TABLET CUEBOARD ?
C              TYPE Y(YES) OR N(NO)"
      CALL MESSAGE(1,NANS)
      IF(NANS.EQ.1HY) CALL DRAWCUE
C
C   MESSAGE - "DO YOU WISH TO POSITION THE GRAPHIC TABLET CUE BOARD ?
C              TYPE Y(YES) OR N(NO)"
  480 CALL MESSAGE(10,NANS)
```

```
      IF(NANS.EQ.1HN) GO TO 500
      CALL SETCUE
      GO TO 600
  500 JCUE=1
C
C   MESSAGE - "INPUT CUE CODES FROM TERMINAL KEYBOARD."
      CALL MESSAGE(8,NDUM)
  600 CALL CUEBRD
C
C   FORMATS
 1000 FORMAT(I5)
 2000 FORMAT(F5.2)
 2100 FORMAT(I5)
 2200 FORMAT(R1,I2,2X,20R1)
      END
```

```
CCAPT
      SUBROUTINE CAPTION
      COMMON
     */BLK/
     *  IHORZ(4),
     *  IVERT(4),
     *  IYSET,
     *  MAXCOL,
     *  MAXIX,
     *  MAXIY,
     *  MAXROW,
     *  MAXSR,
     *  NF
      COMMON
     */BLKCAPT/
     *  MAXLINE,
     *  NCAPT(85,10)
      COMMON
     */BLKJ/
     *  JCUE,
     *  JDRAW,
     *  JMESS,
     *  JSCR,
     *  JSTAG,
     *  JTITLE
   100 ISIZE=3
       NLINE=0
       IY=NF*775
C
C    THE FOLLOWING STATEMENT CLEARS THE SCREEN, NO MESSAGE IS WRITTEN.
       CALL MESSAGE(0,NDUM)
C
C    MESSAGE - "INPUT TITLE LINES. CHARACTER SIZE IS PRESET TO 3."
       CALL MESSAGE(52,NDUM)
   120 NLINE=NLINE + 1
C
C    MESSAGE - "INPUT CHARACTER SIZE." EOF DEFAULT IS PREVIOUS CHARACTER SIZE."
       CALL MESSAGE(47,NDUM)
       CALL ANMODE
       READ(3,1100) ISIZEA
       IF(EOF(3).EQ.0) ISIZE=ISIZEA
C
C    THE FOLLOWING STATEMENT SKIPS MESSAGE LINES, NO MESSAGE IS WRITTEN.
       CALL MESSAGE(-1,NDUM)
C
C    MESSAGE - "TYPE TITLE LINE."
       CALL MESSAGE(48,NDUM)
   140 CALL ANMODE
```

```
      READ(3,1000) (NCAPT(J,NLINE),J=5,84)
      IF(EOF(3).NE.0) GO TO 300
C
C   THE FOLLOWING STATEMENT SKIPS MESSAGE LINES, NO MESSAGE IS WRITTEN.
      CALL MESSAGE(-1,NDUM)
      NCHARS=80
      DO 200 J=1,80
      IF(NCAPT(NCHARS+4,NLINE).NE.55B) GO TO 220
      NCHARS=NCHARS - 1
  200 CONTINUE
  220 J2=NCHARS + 4
      CALL ANMODE
      WRITE(4,1000) (NCAPT(J,NLINE),J=5,J2)
C
C   THE FOLLOWING STATEMENT SKIPS MESSAGE LINES, NO MESSAGE IS WRITTEN.
      CALL MESSAGE(-1,NDUM)
C
C MESSAGE - "IS THIS LINE CORRECT? TYPE Y(YES) OR N(NO)"
      CALL MESSAGE(49,NANS)
      IF(NANS.EQ.1HN) GO TO 260
      NCAPT(1,NLINE)=(MAXIX - NCHARS*IHORZ(ISIZE))/2
      IY=IY - IVERT(ISIZE)
      IF(NLINE.EQ.1) GO TO 240
      ISIZE1=NCAPT(3,NLINE-1)
      IY=IY - 0.125*(IVERT(ISIZE) + IVERT(ISIZE1))
  240 NCAPT(2,NLINE)=IY
      NCAPT(3,NLINE)=ISIZE
      NCAPT(4,NLINE)=NCHARS
      GO TO 120
C
C   MESSAGE - "CHANGE CHARACTER SIZE, IF DESIRED."
  260 CALL MESSAGE(50,NDUM)
      CALL ANMODE
      READ(3,1100) ISIZEA
      IF(EOF(3).EQ.0) ISIZE=ISIZEA
C
C   THE FOLLOWING STATEMENT SKIPS MESSAGE LINES, NO MESSAGE IS WRITTEN.
      CALL MESSAGE(-1,NDUM)
C
C   MESSAGE - "RETYPE LAST LINE"
      CALL MESSAGE(51,NDUM)
      GO TO 140
C
C   THE FOLLOWING STATEMENT CLEARS THE SCREEN, NO MESSAGE IS WRITTEN.
  300 CALL MESSAGE(0,NDUM)
      IYSET=IY - 1*IVERT(ISIZE)
      MAXLINE=NLINE
      ENTRY WTITLE
```

```
      IF(MAXLINE.EQ.0) GO TO 100
      IF(JDRAW.EQ.0 .OR. MAXIY.EQ.IYSET) GO TO 320
      MAXIY=IYSET
      CALL SETSCR
  320 NLINE=0
  340 NLINE=NLINE + 1
      ISIZE=NCAPT(3,NLINE)
      NCHARS=NCAPT(4,NLINE)
      CALL CHRSIZ(ISIZE)
      IX=NCAPT(1,NLINE)
      IY=NCAPT(2,NLINE)
      CALL MCVABS(IX,IY)
      J2=NCHARS + 4
      CALL AMMODE
      WRITE(4,1000) (NCAPT(J,NLINE),J=5,J2)
      IF(NLINE.LT.MAXLINE) GO TO 340
      RETURN
C
C   FORMATS
 1000 FORMAT(80R1)
 1100 FORMAT(I1)
      END
```

```
CCUEBRD
      SUBROUTINE CUEBRD
   40 NTHBLK=0
      CALL INPUT(X,Y,NTHBLK)
   60 IF(NTHBLK.GT.0) GO TO 80
      CALL MESSAGE(6,NDUM)
      GO TO 40
   80 MAXBLK= 40
      IF(NTHBLK.GT.MAXBLK) GO TO 9999
      GO TO (
     *100,105,110,115,120,125,130,135,140,145,150,155,160,165,170,175,
     *9999,180,185,9999,190,195,200,205,210,215,220,225,230,9999,9999,
     *9999,9999,9999,9999,9999,9999,9999,9999,235
     *) NTHBLK
  100 CALL MESFLIP
      GO TO 40
  105 CALL CUEFLIP
      GO TO 40
  110 CALL DRAWCUE
      GO TO 40
  115 CALL SETCUE
      GO TO 40
  120 CALL SETTO
      GO TO 40
  125 CALL SETSCR
      GO TO 40
  130 CALL BORDER
      GO TO 40
  135 CALL AXIS
      GO TO 40
  140 CALL MESSAGE(0,NDUM)
      GO TO 40
  145 CALL HDCOPY
      GO TO 40
  150 CALL NEWLINE
      GO TO 40
  155 CALL SPLINE(NTHBLK),RETURNS(60)
      GO TO 40
  160 CALL LINEAR(NTHBLK),RETURNS(60)
      GO TO 40
  165 CALL VIEWALL
      GO TO 40
  170 CALL VIEW
      GO TO 40
  175 CALL SYMMET
      GO TO 40
  180 CALL CAPTION
      GO TO 40
```

```
  185 CALL WTITLE
      GO TO 40
  190 CALL SAVE
      GO TO 40
  195 CALL REVIVE
      GO TO 40
  200 CALL ERASED
      GO TO 40
  205 CALL STAGLN
      GO TO 40
  210 CALL DRWSTAG
      GO TO 40
  215 CALL SETKEY
      GO TO 40
  220 CALL KEY
      GO TO 40
  225 CALL ISO
      GO TO 40
  230 CALL DRAWISO
      GO TO 40
  235 STOP"REQUESTED FROM CUE BOARD"
      GO TO 40
C
C   MESSAGE - BLANK CUE REQUESTED. TRY AGAIN.
 9999 CALL MESSAGE(17,NUM)
      GO TO 40
      END
```

```
CORWISC
      SUBROUTINE DRAWISC
      COMMON
     */BLKISC/
     *  ANGLE(20),
     *  ARCMAX(20),
     *  NTITLE(8),
     *  MAXLINE,
     *  RADIUS(20,20),
     *  VAR(20),
     *  XISC(60,20),
     *  YISC(60,20)
      COMMON
     */BLKPTS/
     *  ARRAY(500,2),
     *  ASPL(100,20),
     *  NCODE(500),
     *  NPTS,
     *  NSPLS(500),
     *  NSWITCH,
     *  NTHSPL,
     *  NVPTS,
     *  UNIT
       DO 300 NTHLINE=1,MAXLINE
       ARCLEN=0.0
   200 X=SPLNC1(1,XISC(1,NTHLINE),ARCLEN)
       Y=SPLNC1(1,YISC(1,NTHLINE),ARCLEN)
       IF(ARCLEN.EQ.0.0) CALL MOVEA(X,Y)
       CALL DRAWA(X,Y)
       ARCLEN=ARCLEN + UNIT
       IF(ARCLEN.LT.ARCMAX(NTHLINE)) GO TO 200
   300 CONTINUE
       RETURN
       END
```

```
CDRWCUE
      SUBROUTINE DRAWCUE
      COMMON
     */ELK/
     *  IHORZ(4),
     *  IVERT(4),
     *  IYSET,
     *  MAXCOL,
     *  MAXIX,
     *  MAXIY,
     *  MAXROW,
     *  MAXSR,
     *  NF
      COMMON
     */ELKCUE/
     *  IHEIGHT,
     *  ILENGTH,
     *  ISIDE,
     *  IUQDELT,
     *  IXORIG,
     *  IUQ1,
     *  IVQDELT,
     *  IYORIG,
     *  IVQ1,
     *  LEGEND(15,15,20),
     *  NCHARQ,
     *  PHIQ,
     *  SIDEQ
      DIMENSION
     *  NWORDS(20)
C
C   MESSAGE - "IS HARDCOPIER ON AND SUFFICIENTLY WARM ? TYPE Y(YES) OR N(NO)
      CALL MESSAGE(2,NANS)
C
C   MESSAGE - "TURN ON HARDCOPIER, WHEN WARM TYPE Y."
      IF(NANS.EQ.1HN) CALL MESSAGE(3,NDUM)
C
C   THE FOLLOWING STATEMENT CLEARS THE SCREEN, NO MESSAGE IS WRITTEN.
  200 CALL MESSAGE(0,NDUM)
      CALL MOVABS(IXORIG,IYORIG)
      CALL MOVREL(-ILENGTH/2,-IHEIGHT/2)
      CALL DRWREL(0,IHEIGHT)
      DO 300 NCOL=1,MAXCOL
      CALL MOVREL(ISIDE,-IHEIGHT)
      CALL DRWREL(0,IHEIGHT)
  300 CONTINUE
      CALL DRWREL(-ILENGTH,0)
      DO 360 NROW=1,MAXROW
```

```
      CALL MOVREL(ILENGTH,-ISIDE)
      CALL DRWREL(-ILENGTH,0)
  360 CONTINUE
      CALL MOVREL(ILENGTH,IHEIGHT)
      DO 600 NROW=1,MAXROW
      CALL MOVREL(-ILENGTH,-ISIDE/2)
      DO 500 NCOL=1,MAXCOL
      CALL SEELOC(IX,IY)
      NCHAR=0
      DO 400 N=1,NCHARQ
      IF(LEGEND(NROW,NCOL,N).EQ.55B) GO TO 420
      NWORDS(N)=LEGEND(NROW,NCOL,N)
      NCHAR=N
  400 CONTINUE
  420 IF(NCHAR.LT.1) GO TO 480
      DO 440 I=1,4
      ISIZE=I
      IXCAPT=NCHAR*IHORZ(ISIZE)
      IF(IXCAPT.LT.ISIDE) GO TO 460
  440 CONTINUE
C
C   MESSAGE - "CAPTION TO LONG FOR CUE BLOCK"
      CALL MESSAGE(16,NDUM)
  460 CALL MOVREL((ISIDE-IXCAPT)/2,-IVERT(ISIZE)/2)
      CALL CHRSIZ(ISIZE)
      CALL ANMODE
      WRITE(4,2000) (NWORDS(N),N=1,NCHAR)
      CALL MOVABS(IX,IY)
      CALL MOVREL((ISIDE-3*IHORZ(3))/2,((ISIDE/2)-IVERT(3)))
      CALL CHRSIZ(3)
      CALL ANMODE
      WRITE(4,2100) NROW,NCOL
      CALL MOVABS(IX,IY)
  480 CALL MOVREL(ISIDE,0)
  500 CONTINUE
      CALL MOVREL(0,-ISIDE/2)
  600 CONTINUE
      CALL HDCOPY
C
C   MESSAGE - "IS COPY SATISFACTORY ? TYPE Y(YES) OR N(NO)"
      CALL MESSAGE(4,NANS)
      IF(NANS.EQ.1HN) GO TO 200
C
      RETURN
C
C   FORMATS
 2000 FORMAT(20R1)
 2100 FORMAT(R1,I2.2)
```

END

```
CERASE
      SUBROUTINE ERASE
      COMMON
     */BLKPTS/
     *  ARRAY(500,2),
     *  ASPL(100,20),
     *  NCODE(500),
     *  NPTS,
     *  NSPLS(500),
     *  NSWITCH,
     *  NTHSPL,
     *  NVPTS,
     *  UNIT
      ENTRY ERASED
C
C   MESSAGE - "ERASE DRAWING ? TYPE Y(YES) OR N(NO)"
      CALL MESSAGE(36,NANS)
      IF(NANS.EQ.1HN) RETURN
      DO 100 N=1,3000
      ARRAY(N)=0.0
  100 CONTINUE
      DO 200 N=1,1004
      NCODE(N)=0
  200 CONTINUE
      RETURN
      END
```

```
CFIT
      SUBROUTINE FIT(NFIRST)
      COMMON
     */BLKPTS/
     *  ARRAY(500,2),
     *  ASPL(100,20),
     *  NCODE(500),
     *  NPTS,
     *  NSPLS(500),
     *  NSWITCH,
     *  NTHSPL,
     *  NVPTS,
     *  UNIT
      N1=NFIRST
      N2=NPTS
      DO 100 N=N1,N2
      XSPL=ARRAY(N,1)
      YSPL=ARRAY(N,2)
      CALL UPDATE(ASPL(1,NTHSPL),XSPL,YSPL)
  100 CONTINUE
      RETURN
      END
```

```
CINPUT
      SUBROUTINE INPUT(X,Y,NTHBLK)
      COMMON
     */BLK/
     *  IHORZ(4),
     *  IVERT(4),
     *  IYSET,
     *  MAXCOL,
     *  MAXIX,
     *  MAXIY,
     *  MAXROW,
     *  MAXSR,
     *  NF
      COMMON
     */BLKCUE/
     *  IHEIGHT,
     *  ILENGTH,
     *  ISIDE,
     *  IUQDELT,
     *  IXORIG,
     *  IUQ1,
     *  IVQDELT,
     *  IYORIG,
     *  IVQ1,
     *  LEGEND(15,15,20),
     *  NCHARQ,
     *  PHIQ,
     *  SIDEQ
      COMMON
     */BLKJ/
     *  JCUE,
     *  JDRAW,
     *  JMESS,
     *  JSCR,
     *  JSTAG,
     *  JTITLE
      COMMON
     */BLKSCAL/
     *  LENSCAL,
     *  NEXP,
     *  SCALAST,
     *  SCALES,
     *  SCALET,
     *  SCALSIZ
      COMMON
     */BLKSD/
     *  IXRANGE,
     *  IXSTART,
```

```
     *  IX1,
     *  IYRANGE,
     *  IYSTART,
     *  IY1,
     *  NUNIT(20),
     *  PHID,
     *  RATIOD,
     *  XRANGE,
     *  XSTART,
     *  XD,
     *  YRANGE,
     *  YSTART,
     *  YD
      COMMON
     */BLKTD/
     *  IUDRW1,
     *  IUDRW2,
     *  IUD,
     *  IVDRW1,
     *  IVDRW2,
     *  IVD
      DIMENSION
     *  ICHARS(3)
      IF(JCUE.EQ.1) GO TO 400
  100 CALL BELL
      CALL ONEPNT(IU,IV)
      IF(JDRAW.EQ.0 .OR. NTHBLK.EQ.0) GO TO 200
      IF( IU.LT.IUDRW1 .OR. IU.GT.IUDRW2
     *    .OR.
     *    IV.LT.IVDRW1 .OR. IV.GT.IVDRW2 ) GO TO 200
      IUDELT=IU - IUD
      IVDELT=IV - IVD
      X=SCALET*(IUDELT*COS(PHID) + IVDELT*SIN(PHID))
      Y=SCALET*(IVDELT*COS(PHID) - IUDELT*SIN(PHID))
      NTHBLK=-1
      RETURN
  200 IF(IUQ1.GT.0 .AND. IVQ1.GT.0) GO TO 240
C
C   MESSAGE - "GRAPHIC TABLET CUE BOARD HAS NOT BEEN POSITIONED."
      CALL MESSAGE(37,NDUM)
C
C   MESSAGE - "DO YOU WISH A COPY OF THE GRAPHICS TABLET CUEBOARD ?
C              TYPE Y(YES) OR N(NO)"
      CALL MESSAGE(1,NANS)
      IF(NANS.EQ.1HY) CALL DRAWCUE
C
C   MESSAGE - "DO YOU WISH TO POSITION THE GRAPHIC TABLET CUE BOARD ?"
      CALL MESSAGE(10,NANS)
```

```
      IF(NANS.EQ.1HY) GO TO 220
      JCUE=1
C
C   MESSAGE - "INPUT CUE CODES FROM TERMINAL KEYBOARD"
      CALL MESSAGE(2,MDUM)
      GO TO 400
  220 CALL SETCUE
      NTHBLK=-1
      RETURN
  240 IUDELT=IU - IUQ1
      IVDELT=IV - IVQ1
      UPRIME=IUDELT*COS(PHIQ) + IVDELT*SIN(PHIQ)
      VPRIME=IVDELT*COS(PHIQ) - IUDELT*SIN(PHIQ)
      IF( UPRIME.LE.0.0 .OR. UPRIME.GE.IUQDELT
     *    .OR.
     *    VPRIME.LE.0.0 .OR. VPRIME.GE.IVQDELT ) GO TO 300
      NTHCOL=(UPRIME/SIDEQ) + 1
      NTHROW=VPRIME/SIDEQ
      NTHROW=MAXROW - NTHROW
      GO TO 700
C
C   MESSAGE - " POINTER IMPROPERLY POSITIONED, TRY AGAIN."
  300 CALL MESSAGE(6,MDUM)
      GO TO 100
  400 IF(UDRAW.EQ.0 .OR. NTHBLK.EQ.0) GO TO 500
  420 CALL BELL
      CALL SCURSR(ICHAR,IX,IY)
      IF(ICHAR.EQ.81) GO TO 500
C                 "Q"
      IF(IX.LT.IXSTART .OR.
     *   IX.GT.(IXSTART + IXRANGE) .OR.
     *   IY.LT.IYSTART .OR.
     *   IY.GT.(IYSTART + IYRANGE) ) GO TO 420
      IXDELT=IX - IXSTART
      IYDELT=IY - IYSTART
      X=SCALES*IXDELT + XSTART
      Y=SCALES*IYDELT + YSTART
      NTHBLK=-1
      RETURN
  500 NTHROW=0
      NTHCOL=0
      CALL HOME
      CALL BAKSP
      CALL CHRSIZ(3)
      CALL CZAXIS(2)
      CALL BELL
      CALL ANMODE
      READ(3,2000) ICHARS
```

```
      CALL CZAXIS(0)
      NTHROW=ICHARS(1)
      IF(NTHROW.LT.1 .OR. NTHROW.GT.MAXROW) GO TO 600
      IF(ICHARS(2).GE.33B .AND. ICHARS(2).LE.44B)
     *  NTHCOL=ICHARS(2) - 33B
      IF(ICHARS(3).GE.33B .AND. ICHARS(3).LE.44B)
     *  NTHCOL=10*NTHCOL + (ICHARS(3) - 33B)
      IF(NTHCOL.GE.1 .OR. NTHCOL.LE.MAXCOL) GO TO 700
C
C   MESSAGE - "IMPROPER CUE CODE. TRY AGAIN."
  600 CALL MESSAGE(59,NDUM)
      GO TO 500
  700 NTHBLK=(NTHROW - 1)*MAXCOL + NTHCOL
      RETURN
C
C   FORMATS
C
 2000 FORMAT(3R1)
      END
```

```
CISO
      SUBROUTINE ISO
      COMMON
     */BLKISO/
     *  ANGLE(20),
     *  ARCMAX(20),
     *  NTITLE(8),
     *  MAXLINE,
     *  RADIUS(20,20),
     *  VAR(20),
     *  XISO(60,20),
     *  YISO(60,20)
      DIMENSION
     *  X(20),
     *  Y(20)
      PI=3.14159265359
      RD=PI/180.0
C
C   MESSAGE - "INPUT TAPE NUMBER CONTAINING ISO DATA."
      CALL MESSAGE(53,NDUM)
      READ(3,*) NTAPE
      IF(NTAPE.EQ.0) RETURN
      REWIND NTAPE
      READ(NTAPE,1000) NTITLE
      IF(EOF(NTAPE).NE.0) GO TO 990
      WRITE(4,1000) NTITLE
C
C   THE FOLLOWING STATEMENT SKIPS MESSAGE LINES, NO MESSAGE IS WRITTEN.
      CALL MESSAGE(-2,NDUM)
C
C   MESSAGE - "READ THIS DATA ? TYPE Y(YES) OR N(NO)"
      CALL MESSAGE(54,NANS)
      IF(NANS.EQ.1HN) RETURN
      NTHLINE=0
      DO 140 N=1,20
      DO 120 M=1,60
      XISO(M,N)=0.0
      YISO(M,N)=0.0
  120 CONTINUE
  140 CONTINUE
      DO 180 M=1,20
      DO 160 N=1,20
      RADIUS(M,N)=-1.0
  160 CONTINUE
  180 CONTINUE
      READ(NTAPE,1060) XCEN
      IF(EOF(NTAPE).NE.0) GO TO 990
  200 READ(NTAPE,1060) YCEN
```

```
      IF(EOF(NTAPE).NE.0) GO TO 990
      READ(NTAPE,1060) XPT
      IF(EOF(NTAPE).NE.0) GO TO 990
      READ(NTAPE,1060) YPT
      IF(EOF(NTAPE).NE.0) GO TO 990
      XCEN=-XCEN
      YCEN=-YCEN
      XPT=-XPT
      YPT=-YPT
      ARGX=XPT - XCEN
      ARGY=YPT - YCEN
      EPSLN=ATAN2(ARGY,ARGX)
      READ(NTAPE,1040) NVAR
      IF(EOF(NTAPE).NE.0) GO TO 990
      IF(NVAR.LE.20) GO TO 220
C
C   MESSAGE - "NUMBER OF VARIABLES EXCEEDS MAXIMUM LIMIT OF 20."
      CALL MESSAGE(56,NDUM)
      RETURN
  220 READ(NTAPE,1080) (VAR(N),N=1,NVAR)
      IF(EOF(NTAPE).NE.0) GO TO 990
      READ(NTAPE,1040) NANG
      IF(EOF(NTAPE).NE.0) GO TO 990
      IF(NANG.LE.20) GO TO 240
C
C   MESSAGE - "NUMBER OF ANGLES EXCEEDS MAXIMUM LIMIT OF 20."
      CALL MESSAGE(57,NDUM)
      RETURN
  240 READ(NTAPE,1080) (ANGLE(N),N=1,NANG)
      IF(EOF(NTAPE).NE.0) GO TO 990
      DO 280 N=1,NANG
      ANGLE(N)=RAD*ANGLE(N)
  280 CONTINUE
      READ(NTAPE,1000)
      IF(EOF(NTAPE).NE.0) GO TO 990
  300 READ(NTAPE,1120) NTHANG,NTHVAR,RAD
      IF(EOF(NTAPE).NE.0) GO TO 990
      IF(NTHANG.EQ.99 .OR. NTHVAR.EQ.99) GO TO 320
      RADIUS(NTHANG,NTHVAR)=RAD
      GO TO 300
  320 DO 500 NTHVAR=1,NVAR
      ARCLEN=0.0
      NTRIP=0
      NTH=0
      DO 440 NTHANG=1,NANG
      IF(NTRIP.EQ.1) GO TO 360
      IF(RADIUS(NTHANG,NTHVAR).LT.0.0) GO TO 440
      NTRIP=1
```

```
      NTHLINE=NTHLINE+1
      IF(NTHLINE.LE.20) GO TO 380
C
C   MESSAGE - "NUMBER OF ISO CONTOUR LINES EXCEEDS MAXIMUM LIMIT OF 20."
      CALL MESSAGE(58,NDUM)
      RETURN
  360 IF(RADIUS(NTHANG,NTHVAR).GT.0.0) GO TO 380
      ARCMAX(NTHLINE)=ARCLEN
      ARCLEN=0.0
      NTRIP=0
      NTH=0
      GO TO 440
  380 NTH=NTH+1
      R=RADIUS(NTHANG,NTHVAR)
      XPRIM=R*COS(ANGLE(NTHANG))
      YPRIM=R*SIN(ANGLE(NTHANG))
      X(NTHANG)=XPRIM*COS(EPSLN) - YPRIM*SIN(EPSLN) + XCEN
      Y(NTHANG)=XPRIM*SIN(EPSLN) + YPRIM*COS(EPSLN) + YCEN
      IF(NTH.EQ.1) GO TO 420
      ARGX=X(NTHANG) - X(NTHANG-1)
      ARGY=Y(NTHANG) - Y(NTHANG-1)
      SEGLEN=SQRT(ARGX**2 + ARGY**2)
      ARCLEN=ARCLEN + SEGLEN
  420 CALL UPDATE(XISO(1,NTHLINE),ARCLEN,X(NTHANG))
      CALL UPDATE(YISO(1,NTHLINE),ARCLEN,Y(NTHANG))
  440 CONTINUE
      IF(RADIUS(NANG,NTHVAR).LT.0.0) GO TO 500
      N2=NANG
      DO 460 NTHANG=1,3
      ARGX=X(NTHANG) - X(N2)
      ARGY=Y(NTHANG) - Y(N2)
      SEGLEN=SQRT(ARGX**2 + ARGY**2)
      ARCLEN=ARCLEN + SEGLEN
      CALL UPDATE(XISO(1,NTHLINE),ARCLEN,X(NTHANG))
      CALL UPDATE(YISO(1,NTHLINE),ARCLEN,Y(NTHANG))
      N2=NTHANG
  460 CONTINUE
      ARCMAX(NTHLINE)=ARCLEN
  500 CONTINUE
      MAXLINE=NTHLINE
      READ(NTAPE,1060) XCEN
      IF(EOF(NTAPE).EQ.0) GO TO 200
      RETURN
C
C   MESSAGE - EOF ENCOUNTERED READING ISO DATA TAPE."
  990 CALL MESSAGE(55,NDUM)
      RETURN
C
```

```
C  FORMATS
 1000 FORMAT(8A10)
 1040 FORMAT(20X,I2)
 1060 FORMAT(20X,6G12.5)
 1080 FORMAT(6G12.5)
 1120 FORMAT(I2,8X,I2,8X,G12.5)
      END
```

```
CLINES
      SUBROUTINE LINES(NTHBLK),RETURNS(NS60)
      COMMON
     */BLKJ/
     *  JCUE,
     *  JDRAW,
     *  JMESS,
     *  JSCR,
     *  JSTAG,
     *  JTITLE
      COMMON
     */BLKPTS/
     *  ARRAY(500,2),
     *  ASPL(100,20),
     *  NCODE(500),
     *  NPTS,
     *  NSPLS(500),
     *  NSWITCH,
     *  NTHSPL,
     *  NVPTS,
     *  JNIT
      ENTRY NEWLINE
   60 IF(JDRAW.NE.0) GO TO 80
C
C    MESSAGE - "DRAWING DIMENSIONS HAVE NOT BEEN SPECIFIED."
      CALL MESSAGE(23,NDUM)
      RETURN
   80 IF(NSWITCH.EQ.1) GO TO 100
      NPTS=NPTS + 1
      NTHPT=NPTS
      NSWITCH=1
      NCODE(NTHPT)=0
C
C    MESSAGE - "NEW LINE SEGMENT STARTED SELECT LINEAR OR SPLINE."
  100 CALL MESSAGE(29,NDUM)
      RETURN
      ENTRY LINEAR
      IF(NPTS.EQ.0) GO TO 60
      NTHPT=NPTS
      NPLUS=0
      GO TO 200
      ENTRY SPLINE
      IF(NPTS.EQ.0) GO TO 60
      NTHSPL=NTHSPL + 1
      IF(NTHSPL.LE.20) GO TO 120
C
C    MESSAGE - "MAXIMUM NUMBER OF SPLINES HAS BEEN EXCEEDED."
      CALL MESSAGE(31,NDUM)
```

```
      GO TO 400
  120 NFIRST=NPTS
      NSPLS(NFIRST)=NTHSPL
      NPLUS=4
  200 CALL INPUT(X,Y,NTHBLK)
      IF(NTHBLK.NE.-1) GO TO 260
      IF(NCODE(NTHPT).EQ.0 .AND. NSWITCH.EQ.1) GO TO 240
      IF(NPLUS.NE.4 .OR. X.GT.ARRAY(NTHPT,1)) GO TO 220
C
C   MESSAGE - "VALUES OF X MUST BE INCREASING FOR A SPLINE FIT."
      CALL MESSAGE(35,NOUM)
      GO TO 360
  220 NTHPT=NTHPT + 1
      IF(NTHPT.LE.500) GO TO 230
C
C   MESSAGE - "MAXIMUM NUMBER OF POINTS HAS BEEN EXCEEDED."
      CALL MESSAGE(30,NOUM)
      GO TO 400
  230 NCODE(NTHPT)=1 + NPLUS
  240 ARRAY(NTHPT,1)=X
      ARRAY(NTHPT,2)=Y
      NSWITCH=0
      GO TO 200
  260 IF(NCODE(NTHPT).GT.0) GO TO 300
      IF(NCODE(NTHPT).EQ.0) NTHPT=NTHPT - 1
      IF(NPLUS.EQ.4) GO TO 380
      GO TO 400
  300 IF(NCODE(NTHPT).EQ.5 .AND. (NTHPT - NFIRST).LT.2) GO TO 340
      NCODE(NTHPT)=-NCODE(NTHPT)
      NPTS=NTHPT
      IF(NCODE(NTHPT).EQ.-5) CALL FIT(NFIRST)
      RETURN NS60
C
C   MESSAGE - "INSUFFICIENT NUMBER OF POINTS FOR SPLINE CURVE."
  340 CALL MESSAGE(25,NOUM)
C
C   MESSAGE - "DO YOU WISH TO ABORT THIS LINE SEGMENT ? TYPE Y(YES) OR N(NO)"
  360 CALL MESSAGE(26,NANS)
      IF(NANS.EQ.1HN) GO TO 420
  380 NTHSPL=NTHSPL - 1
      NTHPT=NFIRST
C
C   MESSAGE - "CURRENT LINE SEGMENT ABORTED."
  400 CALL MESSAGE(27,NOUM)
      RETURN NS60
C
C   MESSAGE - "INPUT ADDITIONAL POINT(S)."
  420 CALL MESSAGE(29,NOUM)
```

```
GO TO 200
END
```

```
CMESS
      SUBROUTINE MESSAGE(NUM,NANS)
C   THIS SUBROUTINE WRITES MESSAGES TO THE SCREEN
      COMMON
     */ELK/
     *  IHORZ(4),
     *  IVERT(4),
     *  IYSET,
     *  MAXCOL,
     *  MAXIX,
     *  MAXIY,
     *  MAXROW,
     *  MAXSR,
     *  NF
      COMMON
     */ELKJ/
     *  JCODE,
     *  JDRAW,
     *  JMESS,
     *  JSCR,
     *  JSTAG,
     *  JTITLE
      DIMENSION
     *  NCHARS(8,60)
      DATA
     *  (NCHARS=480*55B),
     *  (NTOTAL=60),
     *  (N55=10H          )
      ISIZE=3
      CALL CHRSIZ(ISIZE)
      IF(NUM.LT.1000) GO TO 200
      NLINE=2*IVERT(ISIZE)
C
C   SETTING THE CHARACTER SIZE AND LINE SPACEING FOR MESSAGES.
C
C   THIS SECTION READS THE MESSAGES STORED ON TAPE1
      N55=N55.AND..NOT.MASK(6)
      DO 120 I=1,NTOTAL
      READ(1,1000) NTH,(NCHARS(J,NTH),J=1,8)
      IF(EOF(1).NE.0) RETURN
  120 CONTINUE
      STOP"EOF NOT ENCOUNTERED READING MESSAGES FROM TAPE1."
  200 CALL BELL
      IF(NUM.GT.NTOTAL) STOP"MESSAGE ID NUMBER EXCEEDS MATRIX DIMENSION"
      IF(NUM.GT.0) GO TO 240
      IF(NUM.LT.0) GO TO 220
      NLINE=2*IVERT(ISIZE)
      CALL NEWPAG
```

```
      RETURN
  220 NL=-NUM
      NLINE=NLINE + NL*IVERT(ISIZE)
      RETURN
  240 NCHK1=( NCHARS(1,NUM).AND.MASK(6) ) .OR. N55
      NCHK2=( SHIFT(NCHARS(1,NUM),6) .AND. MASK(6) ) .OR. N55
      IF(NCHK1.NE.1HP .AND.
     *   NCHK2.NE.1HS .AND.
     *   JMESS.NE.1) GO TO 300
  260 NLINE=NLINE + IVERT(ISIZE)
      CALL HOME
      CALL MOVREL(0,-NLINE)
      CALL ANMODE
      WRITE(4,9000) NCHARS(1,NUM),(NCHARS(J,NUM),J=2,8)
      IF(NCHK2.EQ.1HS) CALL FINITT(0,0)
  300 IF(NCHK2.NE.1HQ) RETURN
  340 CALL HOME
      CALL BAKSP
      CALL CZAXIS(2)
      CALL BELL
      CALL ANMODE
      READ(3,8000) NANS
      CALL CZAXIS(0)
      IF(NANS.EQ.1HN .OR. NANS.EQ.1HY) RETURN
      GO TO 260
      ENTRY MESFLIP
      JMESS=-JMESS
      RETURN
C
C   FORMATS
 1000 FORMAT(I2,1X,8R10)
 8000 FORMAT(A1)
 9000 FORMAT(R8,7R10)
      END
```

```
      COPTION
            SUBROUTINE OPTION
            COMMON
           */BLK/
           *  IHORZ(4),
           *  IVERT(4),
           *  IYSET,
           *  MAXCOL,
           *  MAXIX,
           *  MAXIY,
           *  MAXROW,
           *  MAXSR,
           *  NF
            COMMON
           */BLKJ/
           *  JCUE,
           *  JDRAW,
           *  JMESS,
           *  JSCR,
           *  JSTAG,
           *  JTITLE
            COMMON
           */BLKSCAL/
           *  LENSCAL,
           *  NEXP,
           *  SCALAST,
           *  SCALES,
           *  SCALET,
           *  SCALSIZ
            COMMON
           */BLKSD/
           *  IXRANGE,
           *  IXSTART,
           *  IX1,
           *  IYRANGE,
           *  IYSTART,
           *  IY1,
           *  NUNIT(20),
           *  PHID,
           *  RATIOD,
           *  XRANGE,
           *  XSTART,
           *  XD,
           *  YRANGE,
           *  YSTART,
           *  YD
      C
      C*************************************************************************
```

```
C
C   CALCULATES AND DRAWS SCALE KEY ON TERMINAL SCREEN
      ENTRY KEY
      IF(JDRAW.NE.1) GO TO 900
  100 IF(JSCR.NE.1) CALL SETSCR
      IF(SCALAST.EQ.SCALES) GO TO 400
      SCALAST=SCALES
      SCALSIZ=10
      NEXP=0
      STEP=0.125*MAXSR*SCALES
  200 IF(STEP.GE.1.0) GO TO 240
      STEP=10.0*STEP
      NEXP=NEXP - 1
      IF(NEXP.GE.-10) GO TO 200
C
C   MESSAGE - "SCREEN SCALE IS LESS THAN 1.0*E-10."
      CALL MESSAGE(42,NDUM)
      RETURN
  240 IF(SCALSIZ.GE.STEP) GO TO 300
      IF(SCALSIZ.GT.101.0) GO TO 260
      SCALSIZ=10.0*SCALSIZ
      GO TO 240
  260 NEXP=NEXP + 1
      STEP=0.1*STEP
      IF(NEXP.LE.7) GO TO 240
C
C   MESSAGE - "SCREEN SCALE IS GREATER THAN 1.0*E10."
      CALL MESSAGE(43,NDUM)
      RETURN
  300 FACTOR=STEP/SCALSIZ
      IF(FACTOR.GT.0.70) GO TO 340
      SCALSIZ=0.5*SCALSIZ
      IF(SCALSIZ.GT.1.0) GO TO 300
      CALL MESSAGE(44,NDUM)
      RETURN
  340 LENSCAL=(SCALSIZ*(10.0**NEXP))/SCALES
  400 IF(IX1.LT.0 .OR. IY1.LT.0) GO TO 600
      IF((IX1 + 1.2*LENSCAL).GT.(IXSTART + IXRANGE))
     *  IX1=IXSTART + IXRANGE - 1.2*LENSCAL
      CALL MOVABS(IX1,IY1)
      CALL DRWREL(LENSCAL,0)
      IY=0.1*LENSCAL
      CALL DRWREL(0,IY)
      IX=0.5*LENSCAL
      CALL MOVREL(-IX,0)
      CALL DRWREL(0,-IY)
      CALL MOVREL(-IX,0)
      CALL DRWREL(0,IY)
```

```
      IY=IY/2
      CALL MOVREL(0,-IY)
      CALL DRWREL(LENSCAL,0)
      CALL MOVABS(IX1,IY1)
      CALL CHRSIZ(3)
      CALL CSIZE(IH,IV)
      IY=IV
      IX=IH/2
      CALL MOVREL(-IX,-IY)
      CALL ANMODE
      I0=0
      I1=SCALSIZ
      WRITE(4,1000) I0
      CALL MOVABS(IX1,IY1)
      IX=LENSCAL - 2*IH
      IF(I1.LT.1000) IX=IX - IH/2
      IF(I1.LT.100) IX=IX - IH/2
      IF(I1.LT.10) IX=IX -IH/2
      IF(NEXP.NE.0) IX=IX - 3*IH
      CALL MOVREL(IX,-IY)
      CALL ANMODE
      WRITE(4,1040) I1
      IF(NEXP.EQ.0) GO TO 44
      CALL MOVABS(IX1,IY1)
      CALL MOVREL(IX,-IY)
      CALL ANMODE
      WRITE(4,1100) "*10E",NEXP
   44 CALL MOVABS(IX1,IY1)
      DO 46 J=1,20
      IF(NUNIT(J).EQ.55B) GO TO 48
      NCHARS=J
   46 CONTINUE
   48 IX=(LENSCAL - NCHARS*IH)/2
      CALL MOVREL(IX,-2*IV)
      CALL ANMODE
      WRITE(4,2000) (NUNIT(J),J=1,NCHARS)
      RETURN
C
C*****************************************************************************
C
C   REQUESTS POSITION OF SCALE KEY
      ENTRY SETKEY
      IF(JDRAW.NE.1) GO TO 900
C
C   MESSAGE - "SET SCALE POSITION."
   60 CALL MESSAGE(45,NDUM)
      N=-1
      CALL INPUT(X,Y,N)
```

```
      IF(N.EQ.-1) GO TO 640
C
C   MESSAGE - "POINTER IMPROPERLY POSITIONED, TRY AGAIN."
      CALL MESSAGE(6,NDUM)
      RETURN
  640 IX1=IXRANGE*(X - XSTART)/XRANGE
      IY1=IYRANGE*(Y - YSTART)/YRANGE
      RETURN
C
C*******************************************************************************
C
C   DRAWS X-Y AXIS ON TERMINAL SCREEN
      ENTRY AXIS
      IF(JDRAW.NE.1) GO TO 900
      CALL MOVEA(XSTART,0.0)
      X=XSTART + XRANGE
      CALL DASHA(X,0.0,777777636)
      CALL MOVEA(0.0,YSTART)
      Y=YSTART + YRANGE
      CALL DASHA(0.0,Y,777777636)
      RETURN
C
C*******************************************************************************
C
C   DRAWS BORDER ON TERMINAL SCREEN
      ENTRY BORDER
      IF(JDRAW.NE.1) GO TO 900
      CALL MOVABS(IXSTART,IYSTART)
      CALL DRWREL(0,IYRANGE)
      CALL DRWREL(IXRANGE,0)
      CALL DRWREL(0,-IYRANGE)
      CALL DRWREL(-IXRANGE,0)
      RETURN
C
C   MESSAGE - "DRAWING DIMENSIONS HAVE NOT BEEN SPECIFIED."
  900 CALL MESSAGE(23,NDUM)
      RETURN
C
C   FORMATS
 1000 FORMAT(I1)
 1040 FORMAT(I4)
 1100 FORMAT(4X,A4,I2)
 2000 FORMAT(20R1)
      END
```

```
CREVIVE
      SUBROUTINE REVIVE
      COMMON
     */BLK/
     *  IHORZ(4),
     *  IVERT(4),
     *  IYSET,
     *  MAXCOL,
     *  MAXIX,
     *  MAXIY,
     *  MAXROW,
     *  MAXSR,
     *  NF
      COMMON
     */BLKJ/
     *  JCUE,
     *  JDRAW,
     *  JMESS,
     *  JSCR,
     *  JSTAG,
     *  JTITLE
      COMMON
     */BLKPTS/
     *  ARRAY(500,2),
     *  ASPL(100,20),
     *  NCODE(500),
     *  NPTS,
     *  NSPLS(500),
     *  NSWITCH,
     *  NTHSPL,
     *  NVPTS,
     *  UNIT
      COMMON
     */BLKSCAL/
     *  LENSCAL,
     *  NEXP,
     *  SCALAST,
     *  SCALES,
     *  SCALET,
     *  SCALSIZ
      COMMON
     */BLKSD/
     *  IXRANGE,
     *  IXSTART,
     *  IX1,
     *  IYRANGE,
     *  IYSTART,
     *  IY1,
```

```
     *  NUNIT(20),
     *  PHID,
     *  RATIOD,
     *  XRANGE,
     *  XSTART,
     *  XO,
     *  YRANGE,
     *  YSTART,
     *  YO
C
C    MESSAGE - "INPUT TAPE NUMBER THAT CONTAINS DRAWING DATA."
      CALL MESSAGE(38,NDUM)
      CALL ANMODE
      READ(3,*) NTAPE
C
C    THE FOLLOWING STATEMENT SKIPS MESSAGE LINES, NO MESSAGE IS WRITTEN.
      CALL MESSAGE(-2,NDUM)
      IF(EOF(3).NE.0 .OR. NTAPE.LE.0) RETURN
      REWIND NTAPE
      READ(NTAPE,1000) IXRANGE,
     *                 IXSTART,
     *                 IYRANGE,
     *                 IYSTART,
     *                 MAXSR  ,
     *                 NADD
      IF(EOF(NTAPE).NE.0) GO TO 400
      READ(NTAPE,1060) NUNIT
      IF(EOF(NTAPE).NE.0) GO TO 400
      READ(NTAPE,1100) PHID   ,
     *                 RATIOD ,
     *                 SCALET ,
     *                 UNIT   ,
     *                 XRANGE ,
     *                 XSTART ,
     *                 XO     ,
     *                 YRANGE ,
     *                 YSTART ,
     *                 YO
      CALL VWINDO(XSTART,XRANGE,YSTART,YRANGE)
      IF(EOF(NTAPE).NE.0) GO TO 400
      READ(NTAPE,1200)
      IF(EOF(NTAPE).NE.0) GO TO 400
      JDRAW=1
      CALL SETSCR
      NFIRST=0
      N1=NPTS + 1
      N2=NPTS + NADD
      DO 300 NTHPT=N1,N2
```

```
      READ(NTAPE,1300) I,NSPLS(NTHPT),NCODE(NTHPT),ARRAY(NTHPT,1),
     *                 ARRAY(NTHPT,2)
      IF(EOF(NTAPE).NE.0) GO TO 400
      NPTS=NTHPT
      IF(NCODE(NTHPT).NE.-5) GO TO 160
      IF(NTHPT - NFIRST.GT.1) GO TO 140
C
C   MESSAGE - "INSUFFICIENT NUMBER OF POINTS FOR SPLINE CURVE."
      CALL MESSAGE(25,NDUM)
      RETURN
  140 CALL FIT(NFIRST)
      NFIRST=0
  160 IF(NSPLS(NTHPT).EQ.0) GO TO 300
      NFIRST=NTHPT
      NTHSPL=NTHSPL+1
      IF(NTHSPL.LE.20) GO TO 180
C
C   MESSAGE - "MAXIMUM NUMBER OF SPLINES HAS BEEN EXCEEDED."
      CALL MESSAGE(31,NDUM)
      RETURN
  180 NSPLS(NTHPT)=NTHSPL
  300 CONTINUE
      RETURN
C
C   MESSAGE - "EOF ENCOUNTERED READING DRAWING DATA TAPE."
  400 CALL MESSAGE(39,NDUM)
      RETURN
C
C   FORMATS
 1000 FORMAT(10X,I10)
 1060 FORMAT(10X,20R1)
 1100 FORMAT(8X,G13.6)
 1200 FORMAT(8A10)
 1300 FORMAT(I3,4X,I2,3X,I3,3X,2G13.6)
      END
```

```
CSAVE
      SUBROUTINE SAVE
      COMMON
     */BLK/
     *  IHORZ(4),
     *  IVERT(4),
     *  IYSET,
     *  MAXCOL,
     *  MAXIX,
     *  MAXIY,
     *  MAXROW,
     *  MAXSR,
     *  NF
      COMMON
     */BLKJ/
     *  JCUE,
     *  JDRAW,
     *  JMESS,
     *  JSCR,
     *  JSTAG,
     *  JTITLE
      COMMON
     */BLKPTS/
     *  ARRAY(500,2),
     *  ASPL(100,20),
     *  NCODE(500),
     *  NPTS,
     *  NSPLS(500),
     *  NSWITCH,
     *  NTHSPL,
     *  NVPTS,
     *  UNIT
      COMMON
     */BLKSCAL/
     *  LENSCAL,
     *  NEXP,
     *  SCALAST,
     *  SCALES,
     *  SCALET,
     *  SCALSIZ
      COMMON
     */BLKSD/
     *  IXRANGE,
     *  IXSTART,
     *  IX1,
     *  IYRANGE,
     *  IYSTART,
     *  IY1,
```

```
     *  NUNIT(20),
     *  PHIO,
     *  RATIOO,
     *  XRANGE,
     *  XSTART,
     *  XO,
     *  YRANGE,
     *  YSTART,
     *  YO
      IF(JDRAW.NE.0) GO TO 80
C
C   MESSAGE - "DRAWING DIMENSIONS HAVE NOT BEEN SPECIFIED."
      CALL MESSAGE(23,NDUM)
      RETURN
C
C   MESSAGE - "INPUT TAPE NUMBER ON WHICH DRAWING DATA IS TO BE SAVED."
   80 CALL MESSAGE(11,NDUM)
      CALL ANMODE
      READ(3,*) NTAPE
C
C   THE FOLLOWING STATEMENT SKIPS MESSAGE LINES, NO MESSAGE IS WRITTEN.
      CALL MESSAGE(-2,NDUM)
      IF(EOF(3).NE.0 .OR. NTAPE.LE.0) RETURN
      CALL ANMODE
      WRITE(NTAPE,1000) "IXRANGE",IXRANGE,
     *                  "IXSTART",IXSTART,
     *                  "IYRANGE",IYRANGE,
     *                  "IYSTART",IYSTART,
     *                  "MAXSR"  ,MAXSR  ,
     *                  "NPTS"   ,NPTS
      WRITE(NTAPE,1060) "NUNIT"  ,NUNIT
      WRITE(NTAPE,1100) "PHIO"   ,PHIO   ,
     *                  "RATIOO" ,RATIOO,
     *                  "SCALET" ,SCALET ,
     *                  "UNIT"   ,UNIT   ,
     *                  "XRANGE" ,XRANGE ,
     *                  "XSTART" ,XSTART ,
     *                  "XO"     ,XO     ,
     *                  "YRANGE" ,YRANGE ,
     *                  "YSTART" ,YSTART ,
     *                  "YO"     ,YO
      WRITE(NTAPE,1200)
      WRITE(NTAPE,1300) ((I,NSPLS(I),NCODE(I),ARRAY(I,1),ARRAY(I,2))
     *                    ,I=1,NPTS)
      RETURN
C
C   FORMATS
 1000 FORMAT(A10,I10)
```

```
1060 FORMAT(A10,2OR1)
1100 FORMAT(A8,G13.6)
1200 FORMAT(*NTHPT NSPLS NCODE   XARRAY        YARRAY*)
1300 FORMAT(I3,4X,I2,3X,I3,3X,2G13.6)
     END
```

```
CSETCUE
      SUBROUTINE SETCUE
      COMMON
     */BLK/
     *  IHORZ(4),
     *  IVERT(4),
     *  IYSET,
     *  MAXCOL,
     *  MAXIX,
     *  MAXIY,
     *  MAXROW,
     *  MAXSR,
     *  NF
      COMMON
     */BLKCUE/
     *  IHEIGHT,
     *  ILENGTH,
     *  ISIDE,
     *  IJQDELT,
     *  IXORIG,
     *  IJQ1,
     *  IVQDELT,
     *  IYORIG,
     *  IVQ1,
     *  LEGEND(15,15,20),
     *  NCHARQ,
     *  PHIQ,
     *  SIDEQ
      COMMON
     */BLKJ/
     *  JCUE,
     *  JDRAW,
     *  JMESS,
     *  JSCR,
     *  JSTAG,
     *  JTITLE
C
C   MESSAGE - "INPUT LOWER LEFT AND LOWER RIGHT CORNER POINTS OF CUE BOARD"
      CALL MESSAGE(7,NDUM)
      CALL BELL
      CALL ONEPNT(IJQ1,IVQ1)
      CALL BELL
      CALL ONEPNT(IJQ2,IVQ2)
      JDELT=IJQ2 - IJQ1
      VDELT=IVQ2 - IVQ1
      PHIQ=ATAN2(VDELT,JDELT)
      IJQDELT=JDELT
      IVQDELT=MAXROW*(JDELT/MAXCOL)
```

AD-A093 311 NAVAL AIR DEVELOPMENT CENTER WARMINSTER PA AIRCRAFT --ETC F/G 1/3
ZDRAFT-A GRAPHITE CODE FOR VTOL AIRCRAFT GROUND FOOTPRINT VISUA--ETC(U)
JAN 80 J J ZANINE, K A GREEN
UNCLASSIFIED NADC-80109-60 NL

```
      SIDEQ=UDELT/MAXCCL
      JCUE=-1
      RETURN
      ENTRY CUEFLIP
      JCUE=-JCUE
      RETURN
C
C   FORMATS
 2000 FORMAT(I5)
 2100 FORMAT(R1,I2,2X,2OR1)
 2200 FORMAT(2OR1)
      END
```

```
CSETSCR
      SUBROUTINE SETSCR
      COMMON
     */BLK/
     *  IHORZ(4),
     *  IVERT(4),
     *  IYSET,
     *  MAXCOL,
     *  MAXIX,
     *  MAXIY,
     *  MAXROW,
     *  MAXSR,
     *  NF
      COMMON
     */BLKJ/
     *  JCUE,
     *  JDRAW,
     *  JMESS,
     *  JSCR,
     *  JSTAG,
     *  JTITLE
      COMMON
     */BLKPTS/
     *  ARRAY(500,2),
     *  ASPL(100,20),
     *  NCODE(500),
     *  NPTS,
     *  NSPLS(500),
     *  NSWITCH,
     *  NTHSPL,
     *  NVPTS,
     *  UNIT
      COMMON
     */BLKSCAL/
     *  LENSCAL,
     *  NEXP,
     *  SCALAST,
     *  SCALES,
     *  SCALET,
     *  SCALSIZ
      COMMON
     */BLKSD/
     *  IXRANGE,
     *  IXSTART,
     *  IX1,
     *  IYRANGE,
     *  IYSTART,
     *  IY1,
```

```
     *  NUNIT(20),
     *  PHID,
     *  RATIOD,
     *  XRANGE,
     *  XSTART,
     *  XO,
     *  YRANGE,
     *  YSTART,
     *  YO)
   60 IF(JDRAW.NE.0) GO TO 80
C
C    MESSAGE - "DRAWING DIMENSIONS HAVE NOT BEEN SPECIFIED."
      CALL MESSAGE(23,NDUM)
      RETURN
C
C    THE FOLLOWING STATEMENT CLEARS THE SCREEN, NO MESSAGE IS WRITTEN.
   80 CALL MESSAGE(0,NDUM)
      WIDTHS=MAXIX
      HEIGHTS=MAXIY
  100 RATIOS=WIDTHS/HEIGHTS
      IF(RATIOD.GT.RATIOS) GO TO 200
      IYSTART=0
      IYRANGE=HEIGHTS
      IXRANGE=IYRANGE*RATIOD
      IXSTART=(WIDTHS - IXRANGE)/2
      GO TO 240
  200 IXRANGE=WIDTHS
      IXSTART=0
      IYRANGE=IXRANGE/RATIOD
      IYSTART=(HEIGHTS - IYRANGE)/2
C
C    SETTING THE SCREEN WINDOW
  240 CALL SWINDO(IXSTART,IXRANGE,IYSTART,IYRANGE)
      SCALES=XRANGE/IXRANGE
C
C    SETTING THE X INCREMENT FOR SPLINE FIT
      UNIT=5*(MAXSR/1024)*SCALES
      JSCR=1
      RETURN
      END
```

```
CSETTD
      SUBROUTINE SETTD
      COMMON
     */BLKJ/
     *  JCUE,
     *  JDRAW,
     *  JMESS,
     *  JSCR,
     *  JSTAG,
     *  JTITLE
      COMMON
     */BLKSCAL/
     *  LENSCAL,
     *  NEXP,
     *  SCALAST,
     *  SCALES,
     *  SCALET,
     *  SCALSIZ
      COMMON
     */BLKSD/
     *  IXRANGE,
     *  IXSTART,
     *  IX1,
     *  IYRANGE,
     *  IYSTART,
     *  IY1,
     *  NUNIT(20),
     *  PHID,
     *  RATIOD,
     *  XRANGE,
     *  XSTART,
     *  XD,
     *  YRANGE,
     *  YSTART,
     *  YD
      COMMON
     */BLKTD/
     *  IUDRW1,
     *  IUDRW2,
     *  IUD,
     *  IVDRW1,
     *  IVDRW2,
     *  IVD
      IF(JDRAW.NE.0) CALL ERASED
C
C   MESSAGE - "PLACE DRAWING ON THE GRAPHICS TABLET."
      CALL MESSAGE(18,NDUM)
C
```

```
C   MESSAGE - "INPUT LOWER LEFT AND UPPER RIGHT CORNERS OF THE DRAWING"
      CALL MESSAGE(19,NDUM)
      CALL BELL
      CALL ONEPNT(IUDRW1,IVDRW1)
      CALL BELL
      CALL ONEPNT(IUDRW2,IVDRW2)
C
C   MESSAGE - "SPECIFY POINT ON DRAWING AND TYPE IN THE (X,Y) COORDINATES."
      CALL MESSAGE(20,NDUM)
      CALL BELL
      CALL ONEPNT(IU0,IV0)
      CALL BELL
      CALL ANMODE
      READ(3,*) X0,Y0
      IF(EOF(3).NE.0) X0=Y0=0.0
C
C   THE FOLLOWING STATEMENT SKIPS MESSAGE LINES, NO MESSAGE IS WRITTEN.
      CALL MESSAGE(-2,NDUM)
C
C   MESSAGE - "SPECIFY ANOTHER POINT AND TYPE IN (X,Y) COORDINATES."
      CALL MESSAGE(21,NDUM)
      CALL BELL
      CALL ONEPNT(IU1,IV1)
      CALL BELL
      CALL ANMODE
      READ(3,*) X1,Y1
C
C   THE FOLLOWING STATEMENT SKIPS MESSAGE LINES, NO MESSAGE IS WRITTEN.
      CALL MESSAGE(-2,NDUM)
C
C   MESSAGE - "TYPE UNIT OF DIMENSIONS (20 CHARACTERS OR LESS)."
      CALL MESSAGE(46,NDUM)
      CALL ANMODE
      READ(3,1000) NUNIT
      DU=IU1 - IU0
      DV=IV1 - IV0
      ETA=ATAN2(DV,DU)
      DX=X1 - X0
      DY=Y1 - Y0
      THETA=ATAN2(DY,DX)
      PHI0=ETA - THETA
      ARG1=DX**2 + DY**2
      ARG2=DU**2 + DV**2
      IF(ARG2.LE.0.0) CALL MESSAGE(22,NDUM)
      SCALET=SQRT(ARG1/ARG2)
C   SETTING THE SCREEN WINDOW ARGUMENTS
      WIDTH0=IUDRW2 - IUDRW1
      HEIGHT0=IVDRW2 - IVDRW1
```

```
      RATIOD=WIDTHD/HEIGHTD
      IUD=IUD - (XD/SCALET)
      IVD=IVD - (YD/SCALET)
C
C   SETTING THE VIRTUAL WINDOW ARGMENTS
  240 XSTART=(IUDRW1 - IUD)*SCALET
      XRANGE=WIDTHD*SCALET
      YSTART=(IVDRW1 - IVD)*SCALET
      YRANGE=HEIGHTD*SCALET
C
C   SETTING THE VIRTUAL WINDOW
      CALL VWINDC(XSTART,XRANGE,YSTART,YRANGE)
      JDRAW=1
      CALL SETSCR
      RETURN
C
C   FORMATS
 1000 FORMAT(2DR1)
      END
```

```
CSTAGLN
      SUBROUTINE STAGLN
      COMMON
     */BLKJ/
     *  JCUE,
     *  JDRAW,
     *  JMESS,
     *  JSCR,
     *  JSTAG,
     *  JTITLE
C
C   MESSAGE - "INPUT TAPE NUMBER THAT CONTAINS STAGNATION LINE DATA."
   80 CALL MESSAGE(40,NDUM)
      CALL ANMODE
      READ(3,*) JSTAG
C
C   THE FOLLOWING STATEMENT SKIPS MESSAGE LINES, NO MESSAGE IS WRITTEN.
      CALL MESSAGE(-1,NDUM)
      RETURN
      ENTRY DRWSTAG
C
C   MESSAGE - "TAPE NUMBER CONTAINING STAGNATION LINE DATA HAS NOT BEEN SPECIFIE
      IF(JSTAG.GT.0) GO TO 100
      CALL MESSAGE(41,NDUM)
      GO TO 80
  100 REWIND JSTAG
      ISIZE=1
      CALL CHRSIZ(ISIZE)
      CALL CSIZE(IH,IV)
      CALL SEEDW(XMIN,XMAX,YMIN,YMAX)
  120 READ(JSTAG,1000) N,XPT,YPT
      IF(EOF(JSTAG).NE.0)GO TO 200
      XPT=-XPT
      YPT=-YPT
      IF(    (XPT.LT.XMIN)
     *   .OR.(XPT.GT.XMAX)
     *   .OR.(YPT.LT.YMIN)
     *   .OR.(YPT.GT.YMAX) ) GO TO 120
      IF(N.GT.0) GO TO 180
      N=-N
      CALL MOVEA(XPT,YPT)
      IX=0.35*IH
      IY=0.30*IV
      CALL MOVREL(-IX,-IY)
      CALL ANMODE
      WRITE(4,1100) N
      GO TO 120
  180 CALL MOVEA(XPT,YPT)
```

```
      NSIZE=10
      NSYM=3
      CALL SYMBOL(NSYM,NSIZE)
      GO TO 120
  200 RETURN
 1000 FORMAT(I5,2F12.6)
 1100 FORMAT(I1)
      END
```

```
CSYMBOL
      SUBROUTINE SYMBOL(NTHSYM,NSIZE)
C
C   THIS SUBROUTINE DRAWS SYMBOLS FOR THE TEKTRONIX
      ISIZE=NSIZE/2
      GO TO (100,200,300) NTHSYM
C
C  THIS DRAWS AN ASTRIK LIKE FIGURE
  100 DO 140 I=1,8
      IDIR=I-1
      CALL INCPLT(1,IDIR,ISIZE)
      IDIR=IDIR+4
      IF(IDIR.GT.7) IDIR=IDIR-8
      CALL INCPLT(0,IDIR,ISIZE)
  140 CONTINUE
      RETURN
C
C   THIS IS A PLUS SIGN WITH OPPOSITE BARS CONNECTED
  200 CALL INCPLT(1,0,ISIZE)
      CALL INCPLT(1,3,ISIZE)
      CALL INCPLT(1,6,NSIZE)
      CALL INCPLT(1,3,ISIZE)
      CALL INCPLT(1,0,ISIZE)
      RETURN
C
C   THIS DRAWS A SQUARE
  300 CALL INCPLT(0,1,ISIZE)
      CALL INCPLT(1,4,NSIZE)
      CALL INCPLT(1,6,NSIZE)
      CALL INCPLT(1,0,NSIZE)
      CALL INCPLT(1,2,NSIZE)
      RETURN
      END
```

```
CSYMMET
      SUBROUTINE SYMMET
      COMMON
     */BLKJ/
     *  JCUE,
     *  JDRAW,
     *  JMESS,
     *  JSCR,
     *  JSTAG,
     *  JTITLE
      COMMON
     */BLKPTS/
     *  ARRAY(500,2),
     *  ASPL(100,20),
     *  NCODE(500),
     *  NPTS,
     *  NSPLS(500),
     *  NSWITCH,
     *  NTHSPL,
     *  NVPTS,
     *  UNIT
      IF(JDRAW.NE.0) GO TO 80
C
C   MESSAGE - "DRAWING DIMENSIONS HAVE NOT BEEN SPECIFIED."
      CALL MESSAGE(23,NDUM)
      RETURN
   80 IF(NPTS.LT.2) RETURN
      NSTOP=NPTS
      NTHPT1=1
      NTHPT2=NPTS + 1
      IF(NTHPT2.LE.500) GO TO 90
C
C   MESSAGE - "MAXIMUM NUMBER OF POINTS HAS BEEN EXCEEDED."
      CALL MESSAGE(30,NDUM)
      RETURN
   90 NCODE(NTHPT2)=0
  100 IF(IABS(NCODE(NTHPT1 + 1)).NE.5) GO TO 140
      NTHSPL=NTHSPL + 1
      IF(NTHSPL.LE.20) GO TO 120
C
C   MESSAGE - "MAXIMUM NUMBER OF SPLINES HAS BEEN EXCEEDED."
      CALL MESSAGE(31,NDUM)
      RETURN
  120 NFIRST=NTHPT2
      IF(NCODE(NTHPT2).NE.0) NFIRST=NFIRST - 1
      NSPLS(NFIRST)=NTHSPL
  140 NPTS=NTHPT2
      ARRAY(NTHPT2,1)=ARRAY(NTHPT1,1)
```

```
      ARRAY(NTHPT2,2)=-ARRAY(NTHPT1,2)
      IF(NCODE(NTHPT2).EQ.-5) CALL FIT(NFIRST)
      NTHPT1=NTHPT1 + 1
      IF(NTHPT1.GT.NSTOP) GO TO 200
      NTHPT2=NTHPT2 + 1
      NCODE(NTHPT2)=NCODE(NTHPT1)
      IF(NCODE(NTHPT2 - 1).EQ.NCODE(NTHPT2) .OR.
     *   NCODE(NTHPT2 - 1).EQ.0             .OR.
     *   NCODE(NTHPT2).LT.0                 )GO TO 140
      GO TO 100
  200 RETURN
      END
```

```
CVIEW
      SUBROUTINE VIEW
      COMMON
     */BLKJ/
     *  JCUE,
     *  JDRAW,
     *  JMESS,
     *  JSCR,
     *  JSTAG,
     *  JTITLE
      COMMON
     */BLKPTS/
     *  ARRAY(500,2),
     *  ASPL(100,20),
     *  NCODE(500),
     *  NPTS,
     *  NSPLS(500),
     *  NSWITCH,
     *  NTHSPL,
     *  NVPTS,
     *  UNIT
      IF(JDRAW.NE.0) GO TO 60
C
C   MESSAGE - "DRAWING DIMENSIONS HAVE NOT BEEN SPECIFIED."
      CALL MESSAGE(23,NDUM)
      RETURN
   60 NTHPT=NVPTS + 1
      GO TO 120
      ENTRY VIEWALL
      IF(JDRAW.NE.0) GO TO 80
C
C   MESSAGE - "DRAWING DIMENSIONS HAVE NOT BEEN SPECIFIED."
      CALL MESSAGE(23,NDUM)
      RETURN
   80 NTHPT=1
  100 NSTART=1
  120 IF(NCODE(NTHPT).NE.0) NTHPT=NTHPT - 1
      X=ARRAY(NTHPT,1)
      Y=ARRAY(NTHPT,2)
      XBEGIN=X
      CALL MOVEA(X,Y)
  200 NTHPT=NTHPT + 1
      IF(NTHPT.GT.NPTS) GO TO 600
      IF(NCODE(NTHPT).NE.0) GO TO 300
      IF(NSTART.EQ.0) GO TO 100
C
C   MESSAGE - "TWO CONSECUTIVE NCODES EQUAL 0."
      CALL MESSAGE(32,NDUM)
```

```
      RETURN
  300 IF(IABS(NCODE(NTHPT)).NE.1) GO TO 400
      X=ARRAY(NTHPT,1)
      Y=ARRAY(NTHPT,2)
      CALL DRAWA(X,Y)
      IF(NCODE(NTHPT).GT.0) GO TO 200
      NSTART=0
      XBEGIN=X
      GO TO 200
  400 IF(IABS(NCODE(NTHPT)).EQ.5) GO TO 420
C
C   MESSAGE - "NCODE IS NOT -5 , -1 , 0 , 1 , 5."
      CALL MESSAGE(33,NDUM)
      RETURN
  420 NTHSPL=NSPLS(NTHPT-1)
  440 NTHPT=NTHPT + 1
      IF(NTHPT.GT.NPTS) GO TO 600
      IF(NCODE(NTHPT).EQ.5) GO TO 440
      IF(NCODE(NTHPT).EQ.-5) GO TO 460
C
C   MESSAGE - "NCODE SHOULD BE EQUAL TO -5."
      CALL MESSAGE(34,NDUM)
      RETURN
  460 XEND=ARRAY(NTHPT,1)
      YEND=ARRAY(NTHPT,2)
      X=XBEGIN
  480 X=X + UNIT
      IF(X.GT.XEND) GO TO 500
      Y=SPLNQ1(1,ASPL(1,NTHSPL),X)
      CALL DRAWA(X,Y)
      GO TO 480
  500 CALL DRAWA(XEND,YEND)
      NSTART=0
      XBEGIN=X
      GO TO 200
  600 NVPTS=NTHPT - 1
      RETURN
      END
```

(B.4) Listing of the "CREATE" Computer Code

The computer code, "CREATE", reads the second file on the "ZDRAFT" input file and writes the "ZDRAFT" subroutine, CUEBRD. This subroutine manages the cueboard inputs and calls the specified subroutine. Hence, additions and revisions to the cueboard can be made by rewriting the input file. The new subroutine can be LIBEDIT into the existing "ZDRAFT". The permanent file containing the fortran compiled version of "CREATE" is named CREL4. A proceedure file to run "CREATE" follows.

```
PCR
GET,XTAPE,CREL4.
SKIPF,XTAPE.
COPYBF,XTAPE,TAPE2.
REWIND,TAPE2.
CREL4,,,TAPE2,,,CUEBRD.
```

The listing of the "CREATE" code follows.

```
CCREATE
      PROGRAM CREATE(INPUT,OUTPUT,TAPE2,TAPE4,TAPE5,TAPE6,
     *  TAPE8=INPUT,TAPE9=OUTPUT)
      DIMENSION
     *  NFORTRN(225,6),
     *  NCHAR(80),
     *  NDUMMY(80),
     *  NTHCOL(225),
     *  NTHROW(225)
      DATA
     *  (NTHCOL=225*0),
     *  (NTHROW=225*0)
      WRITE(6,1400) "CCUEBRD"
      WRITE(6,1000) "SUBROUTINE CUEBRD"
      WRITE(6,1400) "   40 NTHBLK=0"
      WRITE(6,1400) "      CALL INPUT(X,Y,NTHBLK)"
      WRITE(6,1400) "   60 IF(NTHBLK.GT.0) GO TO 80"
      WRITE(6,1000) "CALL MESSAGE(6,NDUM)"
      WRITE(6,1000) "GO TO 40"
      REWIND 2
      READ(2,2000)
      IF(EOF(2).NE.0) STOP"INPUT DATA NOT FOUND ON TAPE2"
      READ(2,2000) MAXROW,MAXCOL,MAXCHAR
      IF(EOF(2).NE.0) STOP"INPUT DATA NOT FOUND ON TAPE2"
      MAXBLK=MAXROW*MAXCOL
      IF(MAXBLK.GT.225) STOP"CUE BOARD SIZE EXCEEDS MATRIX DIMENSION."
      DO 100 I=1,MAXBLK
      READ(2,2100) NTHROW(I),NTHCOL(I),(NFORTRN(I,N),N=1,6)
      IF(EOF(2).NE.0) GO TO 120
      MAX=I
  100 CONTINUE
  120 I=1
      NSTATE=95
      DO 300 NROW=1,MAXROW
      DO 280 NCOL=1,MAXCOL
      IF(NROW.LT.NTHROW(I) .OR. NCOL.LT.NTHCOL(I)) GO TO 260
      NSTATE=NSTATE + 5
      WRITE(5,1100) NSTATE,(NFORTRN(I,N),N=1,6)
      WRITE(5,1000) "GO TO 40"
      WRITE(4,1200) NSTATE
      I=I+1
      IF(I.LE.MAX) GO TO 280
      MAXBLK=MAXCOL*(NROW-1) + NCOL
      GO TO 320
  260 WRITE(4,1200) 9999
  280 CONTINUE
  300 CONTINUE
  320 WRITE(6,1700) "80 MAXBLK=",MAXBLK
```

```
      WRITE(6,1000) "IF(NTHBLK.GT.MAXBLK) GO TO 9999"
      WRITE(6,1000) "GO TO ("
      REWIND 4
      REWIND 5
      J=0
  360 NBLANK=0
      READ(4,1300) (NDUMMY(I),I=1,6)
      IF(EOF(4).NE.0) GO TO 520
      DO 500 I=1,6
      IF(NDUMMY(I).NE.55B) GO TO 400
      NBLANK=NBLANK + 1
      IF(NBLANK.GT.5) GO TO 360
      GO TO 500
  400 J=J+1
      NCHAR(J)=NDUMMY(I)
      IF(J.LT.62 .OR. NCHAR(J).NE.56B) GO TO 500
      WRITE(6,1500) (NCHAR(K),K=1,J)
      J=0
      GO TO 360
  500 CONTINUE
      GO TO 360
  520 IF(J.GT.0) GO TO 540
      WRITE(6,1600) "*9999) NTHBLK"
      GO TO 560
  540 IF(NCHAR(J).EQ.56B) J=J-1
      WRITE(6,1500) (NCHAR(K),K=1,J)
      WRITE(6,1600) "*) NTHBLK"
  560 READ(5,1300) NDUMMY
      IF(EOF(5).NE.0) GO TO 600
      WRITE(6,1300) NDUMMY
      GO TO 560
  600 WRITE(6,1400) "C"
      WRITE(6,1400) "C   MESSAGE - BLANK CUE REQUESTED. TRY AGAIN."
      WRITE(6,1100) 9999,"CALL MESSAGE(17,NDUM)"
      WRITE(6,1000) "GO TO 40"
      WRITE(6,1000) "END"
C
C   FORMATS
 1000 FORMAT(T7,6A10)
 1100 FORMAT(I5,1X,6A10)
 1200 FORMAT(I4,*,*)
 1300 FORMAT(60R1)
 1400 FORMAT(8A10)
 1500 FORMAT(T6,1H*,66R1)
 1600 FORMAT(T6,7A10)
 1700 FORMAT(T4,A10,I3)
 2000 FORMAT(I5)
 2100 FORMAT(R1,I2,22X,6A10)
```

END

THIS PAGE INTENTIONALLY LEFT BLANK.

DISTRIBUTION LIST

Report No. NADC-80109-60

Airtask Nos. A330330D/001C/9W0582D0100
Work Unit No. RB402

	No. of Copies
NAVAIRSYSCOM (AIR-950D)	6
(2 for retention)	
(1 for AIR-330 :R. Brown)	
(1 for AIR-330B:E. Lichtman)	
(1 for AIR-5301:F. Paglianeto)	
(1 for AIR-3200:D. Kirkpatrick)	
NAVAIRPROPCEN	2
(1 for PE42:A. Atkinson)	
(1 for PE43:L. Palcza)	
Office of Naval Research, Arlington, Virginia . .	1
(1 for Dr. R. Whitehead)	
DDC	12